

XOJO

Desktop-Tutorial

2014 Release 1

Xojo, Inc.

Kapitel 1

Einführung

Willkommen bei Xojo, dem einfachsten Weg zur Erstellung von plattformübergreifenden Desktop- und Web-Anwendungen.



Abschnitt 1

Über dieses Desktop-Tutorial

XOJO STARTEN

1. **Laden Sie die Installationsdatei für Ihr Betriebssystem hier herunter: http://www.xojo.com/download**
2. **Führen Sie das Installationsprogramm aus.**
3. **Starten Sie Xojo.**
4. **Wählen Sie in der Projektauswahl Desktop aus und klicken Sie auf OK.**

Dieses Desktop-Tutorial wendet sich an Xojo- und Programmier-Neulinge. Es stellt eine Einführung in die Desktop-Entwicklungsumgebung dar und führt Sie durch die Erstellung einer echten Desktop-Anwendung.

Die Bearbeitungszeit für dieses Tutorial liegt bei etwa einer Stunde.

Aufgemerkt: Wenn Sie bereits über Erfahrungen mit anderen Programmiersprachen verfügen, sollten Sie sich auch das Benutzerhandbuch und die Sprachreferenz anschauen.

Copyright

Warenmarke

Begriffe, die vermutlich als Warenzeichen oder Service Mark eingetragen sind, werden in Großschreibung wiedergegeben, obwohl Xojo, Inc. über ihren Markenzustand keine verbindliche Zusage treffen kann. Die Verwendung dieser Begriffe stellt keine Verletzung der zugehörigen Marken- oder Urheberrechte dar. Xojo, Inc. steht in keiner Verbindung zu den in diesem Buch erwähnten Produkten oder ihren Herstellern.

Anmerkung des Übersetzers:
Zurzeit liegt Xojo nicht lokalisiert vor, das heißt die Programmumgebung erscheint in englischer Sprache. Deshalb werden in diesem Tutorial die englisch erscheinenden Begriffe auch englisch wiedergegeben, bei ihrer Ersterwähnung aber aus Verständnisgründen deutsch übersetzt. Sie können sich über das Glossar dieses Tutorials die Übersetzungen der Begriffe anzeigen lassen.

Bedeutung von Abkürzungen, Zeichen und Schriftarten

Dieses Tutorial verwendet Screenshots der Xojo-Windows-, OS X- und Linux-Versionen. Aufbau des Benutzerinterfaces und Funktionalität sind auf allen Plattformen identisch. Abweichungen in der Darstellung ergeben sich aus den unterschiedlichen graphischen Benutzerinterfaces der Betriebssysteme.

- **Fettschrift** wird sowohl zur Markierung der Einführung eines neuen Begriffs verwendet als auch zur Hervorhebung wichtiger Konzepte.
- Buchtitel wie *Xojo-Benutzerhandbuch* sind kursiv geschrieben.
- Angaben zur Verwendung eines Eintrags aus den Programm-Menüs werden in der Form „Wählen Sie Datei ↠ Neues Projekt“ gegeben. Dieses Beispiel ist die Kurzform von „Wählen Sie ‚Neues Projekt‘ aus dem Datei-Menü.“
- Tastaturkürzel bestehen aus einer Reihe von Tasten, die in der angegebenen Reihenfolge gedrückt und gehalten werden sollten. Unter Windows und Linux wird die STRG-Taste als Steuerungstaste verwendet, unter OS X dient dazu die ⌘ (Befehl)-Taste. Das Kürzel „STRG+O“ oder „⌘-O“ bedeutet also, die STRG-Taste auf Windows oder Linux-Rechnern oder die ⌘-Taste auf OS X gedrückt zu halten und dann zusätzlich die „O“-Taste zu drücken. Heben Sie den Finger von der STRG- bzw. Befehl-Taste erst, wenn Sie die „O“-Taste nicht mehr drücken.
- In Anführungszeichen stehen Begriffe, die Sie eingeben sollten, wie „StartKnopf”.

- Mitunter wird von Ihnen erwartet, Programmbefehle in den Code-Editor einzugeben. Diese werden in grauen Kästen angezeigt:

  ```
  ShowURL(GewählteURL.Text)
  ```

Bitte beachten Sie bei der Codeeingabe folgendes:

- Geben Sie jede vorgegebene Zeile als einzelne Zeile in den Code-Editor ein. Versuchen Sie nicht, mehrere Programmzeilen zu einem Satz zusammenzufassen oder eine lange Zeile mittels Zeilenschaltungen (Return) zu unterteilen.
- Fügen Sie keine weiteren Leerzeichen ein, wenn in den Programmierzeilen im Tutorial keine stehen.
- Sie können die Codezeilen natürlich auch über die Zwischenablage aus dem Tutorial in den Code-Editor kopieren.

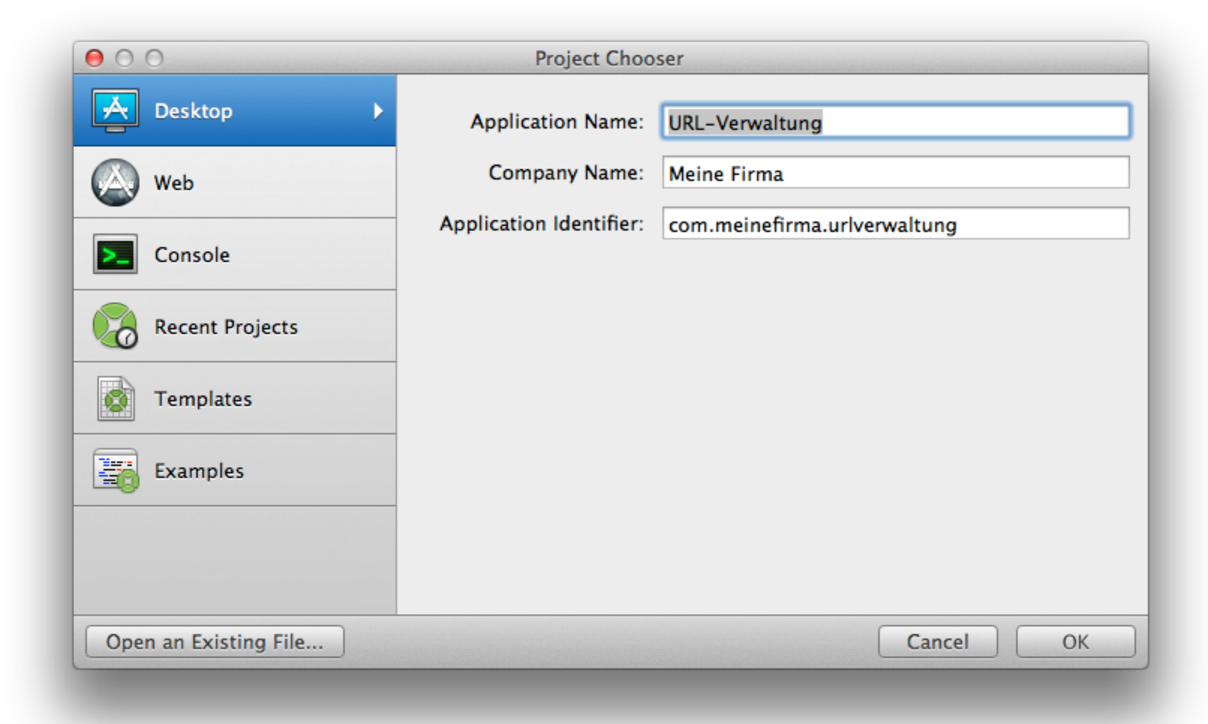
Wenn Sie Ihre Anwendung **kompilieren**, überprüft Xojo zunächst Ihren Code auf Tipp- und Syntaxfehler. Sollte dabei ein Fehler erkannt werden, erscheint eine Fehlerliste zur Überprüfung am Fuß des Xojo-Hauptfensters.

Abschnitt 2

Erste Schritte

Falls Sie das noch nicht getan haben: Starten Sie Xojo.

Abb. 1.1 Das Projektauswahl-Fenster (Project Chooser)



Doppelklicken Sie das Xojo-Programmsymbol, um Xojo zu starten. Nach Abschluss des Ladevorgangs erscheint das Projektauswahl-Fenster.

Xojo erlaubt die Erzeugung dreier verschiedener Anwendungsarten (Desktop, Web und Kommandozeile).

In diesem Tutorial erstellen wir eine Desktop-Applikation. Klicken Sie also auf Desktop.

Sie sehen nun drei Texteingabefelder: Anwendungsname (Application Name), Firmenname (Company Name) und Anwendungs-ID (Application Identifier).

Application Name ist der Name Ihrer Anwendung. Der hier eingegebene Begriff bestimmt den Dateinamen Ihrer Anwendung.

Company Name ist der Name Ihres Unternehmens. Sie können dieses Feld auch leer lassen.

Application Identifier stellt eine eindeutige Kennung Ihrer Anwendung dar. Aus den Eingaben der Felder Application Name und Company Name wird eine Vorgabe für dieses Feld erzeugt. Sie können Sie aber beliebig verändern.

Geben Sie „URL-Verwaltung“ als Anwendungsnamen ein und verändern Sie die Vorgabe für den Company Name ganz nach Lust und Laune oder lassen Sie sie so wie vorgegeben.

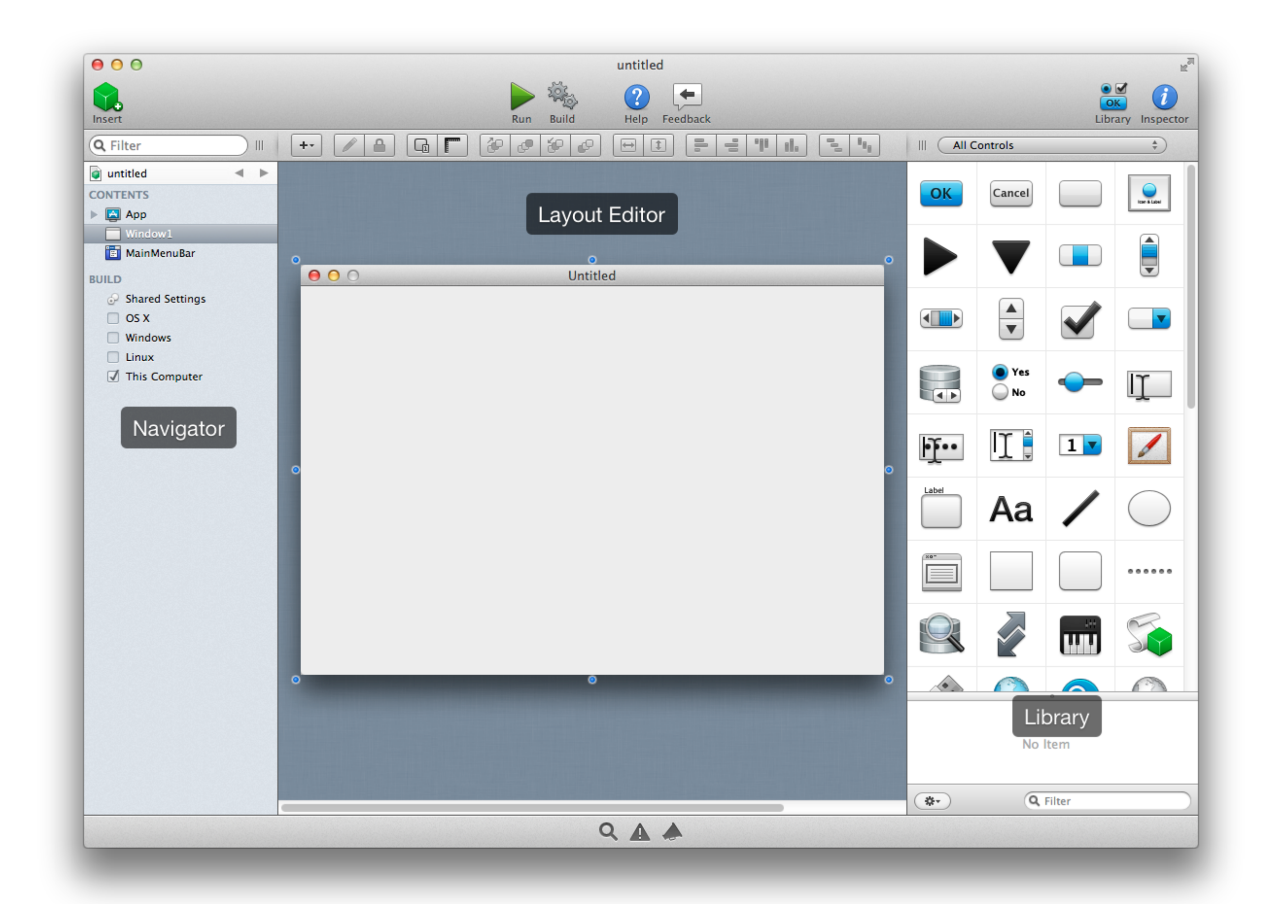
Klicken Sie auf **OK**, um das Hauptfenster von Xojo zu öffnen – den Arbeitsbereich / Workspace), in dem Sie Ihre Anwendung erstellen.

Abschnitt 3

Der Arbeitsbereich (Workspace)

Xojo öffnet den Arbeitsbereich (**Workspace**) mit den für Ihre Anwendungsart passenden Vorgabewerten im **Navigator** und im **Layout-Editor**.

Abb. 1.2 Das Xojo-Hauptfenster



Navigator: Das Panel am linken Bildschirmrand listet alle Bestandteile Ihres Projekts auf. Standardmäßig sehen Sie hier Window1 (das ausgewählt ist), das App-Objekt und das MainMenuBar-Objekt. Mittels des Navigators bewegen Sie sich innerhalb Ihres Projekts.

Layout-Editor: Das mittlere Gebiet des Workspace ist der Layout-Editor. Mit diesem erstellen Sie das Benutzerinterface für die Fenster Ihrer Anwendung. Der Layout-Editor liefert eine Vorschau auf das Erscheinungsbild Ihres Programmfensters. In der Abbildung ist das Fenster leer, weil noch keine Steuerelemente aus der Bibliothek (**Library**) hinzugefügt wurden.

Bibliothek (**Library)**: Das Gebiet am rechten Rand des Hauptfensters stellt die Library dar, in der Sie die Steuer- und Interface-Elemente finden, die Sie zu einem Fenster oder zum Projekt hinzufügen können. Sie gestalten Ihr Fenster, indem Sie die Elemente aus der Library auf dieses hinüberziehen oder indem Sie auf ein Element in der Library doppelklicken.
Sie können das Erscheinungsbild der Elemente in der Library verändern, indem Sie auf das kleine Zahnradsymbol klicken und eine andere Einstellung auswählen.

Aufgemerkt: Sollte die Library nicht sichtbar sein, klicken Sie auf das Library-Symbol in der Werkzeugleiste.

Inspector: In der Abbildung nicht sichtbar ist der Inspector, in dem Sie die Eigenschaften des ausgewählten Steuerelements sehen und verändern können. Der Inspector teilt sich das rechte Panel des Hauptfensters mit der Library.
Sie können sich den Inspector anzeigen lassen, indem Sie auf das Inspector-Symbol in der Werkzeugleiste klicken.
Der Inspector liefert Informationen über das im Navigator oder Editor ausgewählte Element. Das Aussehen des Inspectors wird dem gerade gewählten Element angepasst.
Sie verändern eine Eigenschaft des angezeigten Elements, indem Sie einen Wert in das Texteingabefeld rechts der Feldbezeichnung eingeben.

Die Tutorial-Anwendung

In diesem Tutorial erstellen wir eine Anwendung, die URLs und E-Mail-Adressen verwaltet.

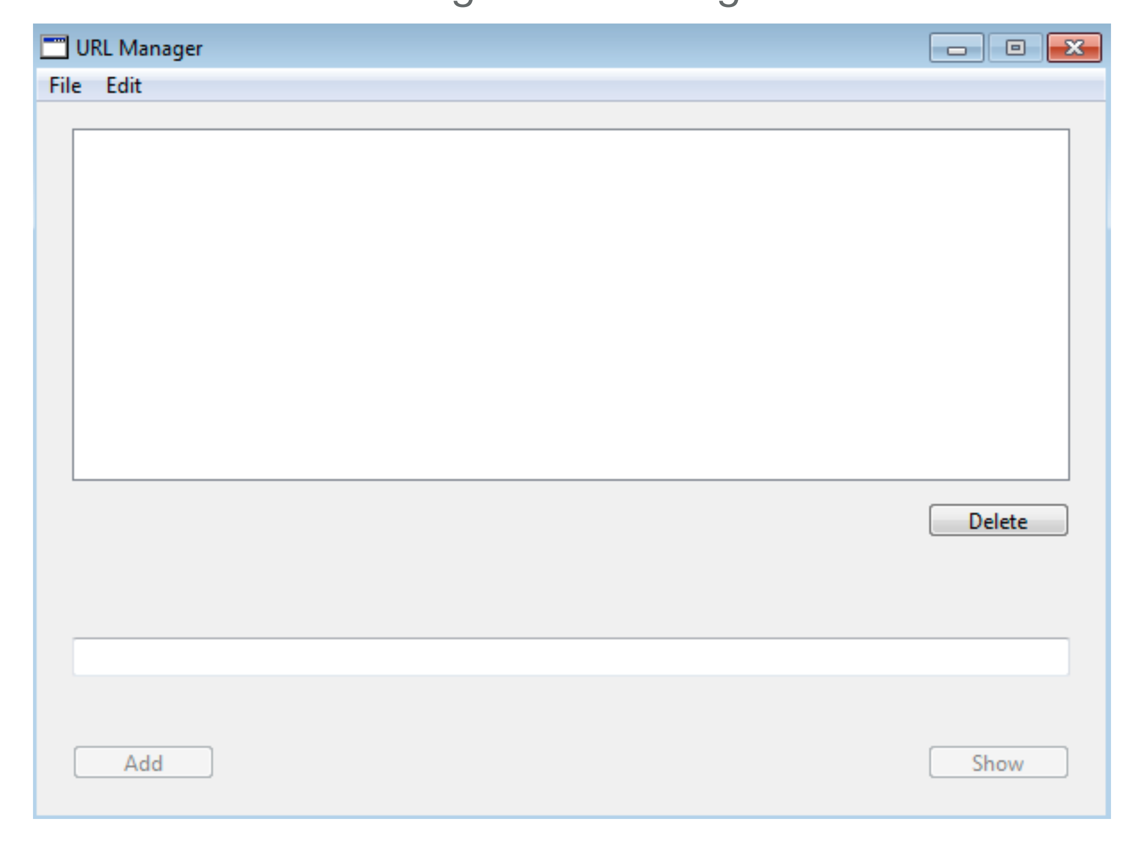
Eine **URL** wird üblicherweise mit „**http**://“ eingeleitet, womit angegeben wird, dass es sich bei der folgenden Adresse um eine Webseite handelt, die angezeigt werden soll.

Steht am Anfang einer URL „**mailto**:“, wird die folgende Adresse stattdessen verwendet, um eine E-Mail über den voreingestellten Mail-Client zu versenden.

URL-Verwaltung

Die URL-Verwaltungs-anwendung nimmt eine URL in einem Textfeld entgegen. Ein Klick auf Anzeigen zeigt diese im voreingestellten Webbrowser an.
Mittels Klick auf Hinzufügen wird die URL in der Liste gespeichert.

Abb. 1.3 URL-Verwaltungs-Anwendung



Um eine URL aus der Liste anzuzeigen, klickt man auf diese und wählt Anzeigen.
Eine URL kann wieder aus der Liste entfernt werden, indem man sie auswählt und auf Löschen klickt.

Die URL-Verwaltung benutzt drei Arten von Steuerelementen:

Listenelement (**List Box**): Ein Steuerelement, das eine scrollbare Liste darstellt. Dies kann sowohl eine einfache Liste als auch eine mehrspaltige Tabelle sein. Scrollen ist sowohl vertikal als auch horizontal möglich.

Textfeld (**Text Field**): Ein Steuerelement, das die Anzeige oder Eingabe einer Textzeile ermöglicht.

Schaltfläche (**Button**): Ein normaler Bildschirmknopf. Häufig benutzt, um eine Programmaktion einzuleiten.

Kapitel 2

Gestaltung des Benutzer-interface

Wir entwerfen die Benutzerschnittstelle.



URL-Liste

Hinzufügen der URL-Liste

Sie sollten Xojo mit geöffnetem Layout-Editor vor sich haben. Falls nicht, lesen Sie bitten Kapitel 1, Abschnitte 3 und 4.

Wir werden dem Fenster nun eine List Box hinzufügen. In dieser List Box werden später die gesammelten URLs angezeigt.

1. Hinzufügen der List Box:

   Klicken Sie in der Library auf die List Box und ziehen Sie sie in die obere linke Ecke des Layout-Editors.

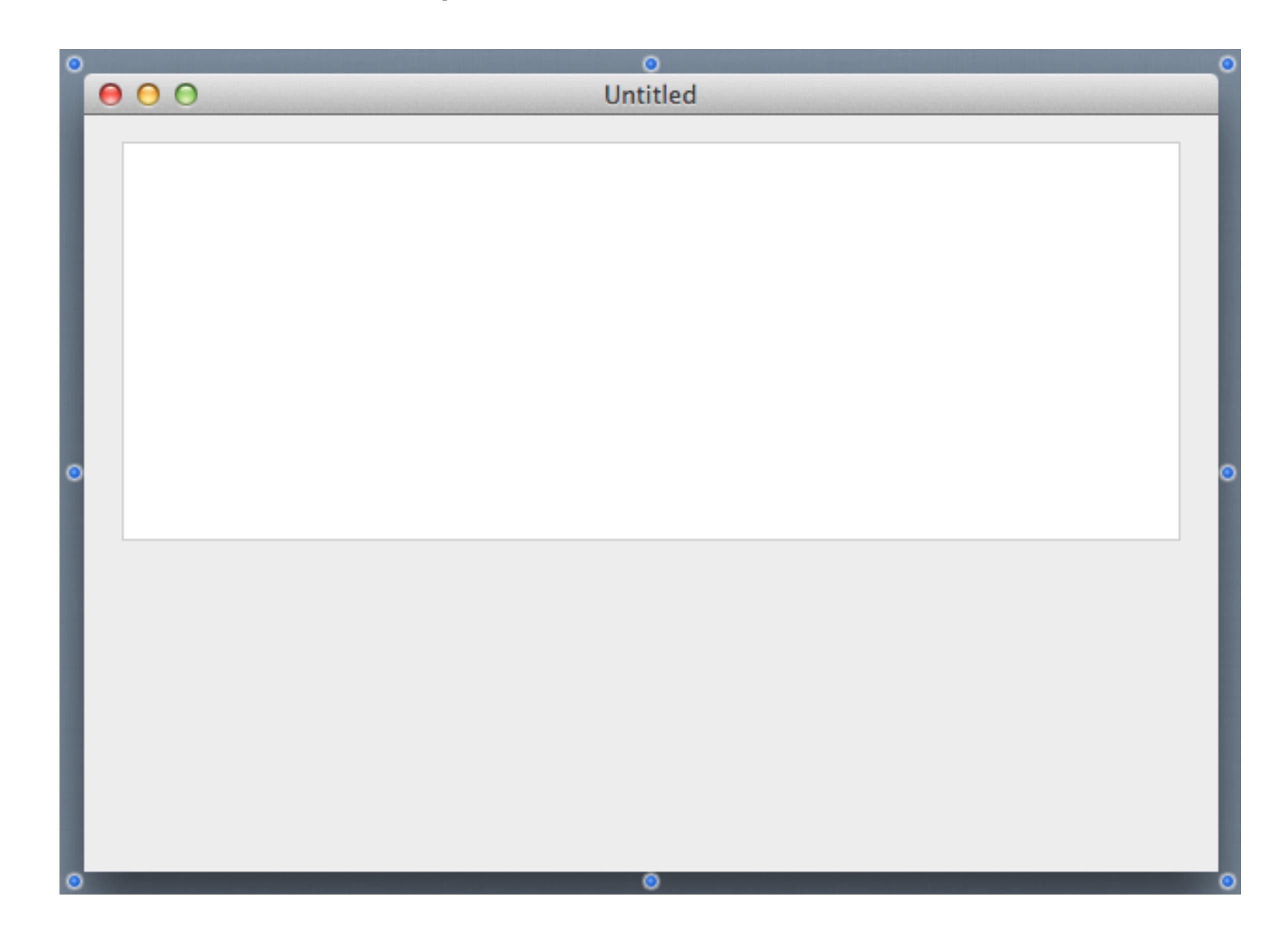
   Wenn Sie den Fensterrändern nahekommen, werden Sie Positionierungshilfslinien bemerken, die Ihnen bei der Platzierung Ihres Elements helfen. Lassen Sie die Maustaste los, wenn Ihnen die Position der List Box innerhalb des Fensters gefällt.

2. Anpassung der List Box-Größe:

   Klicken Sie auf die List Box, damit ihre Anfasser erscheinen. Klicken Sie auf den Anfasser in der rechten unteren Ecke und ziehen Sie bei gedrückter Maustaste, bis die List Box etwa ⅔ des Fensters ausfüllt.

3. Ihr Fenster sollte jetzt ungefähr so aussehen:

Abb. 2.1 Fenster-Layout mit List Box



Schaltflächen (Buttons)

Hinzufügen der Buttons

Jetzt werden wir die Löschen-Schaltfläche hinzufügen. Mittels dieser sollen URLs wieder aus der Liste entfernt werden können.

1. Hinzufügen der Löschen-Schaltfläche:

   Klicken Sie in der Library auf das Steuerelement für allgemeine Schaltflächen (**Generic Button**) und ziehen Sie es in das Fenster unter die rechte untere Ecke der List Box.

   Achten Sie auf die automatischen Hilfslinien, um den Button bündig mit der rechten Kante der Listbox zu positionieren.

2. Hinzufügen der Hinzufügen-Schaltfläche:

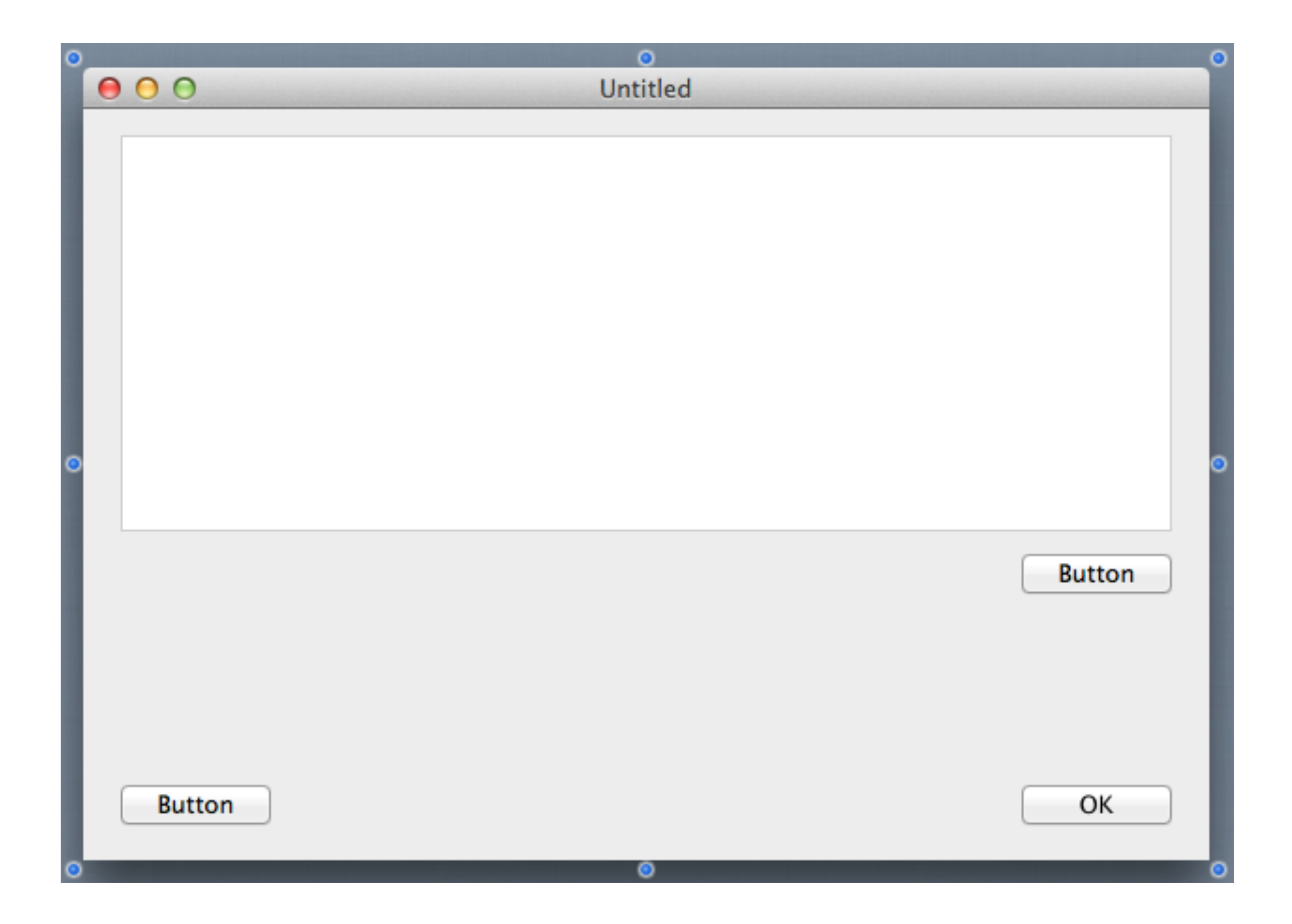
   Klicken Sie in der Library auf das Generic Button-Steuerelement und ziehen Sie es in die Nähe der linken unteren Fensterecke. Die Hilfslinien helfen Ihnen wieder dabei.

3. Hinzufügen der Anzeigen-Schaltfläche:

   Ziehen Sie ein ein Vorgabe-Button-Steuerelement (**Default Button**) aus der Library und positionieren Sie es in der rechten unteren Ecke des Fensters

4. Ihr Fenster sollte nun ungefähr so aussehen:

Abb. 2.2 Fenster-Layout mit Schaltflächen



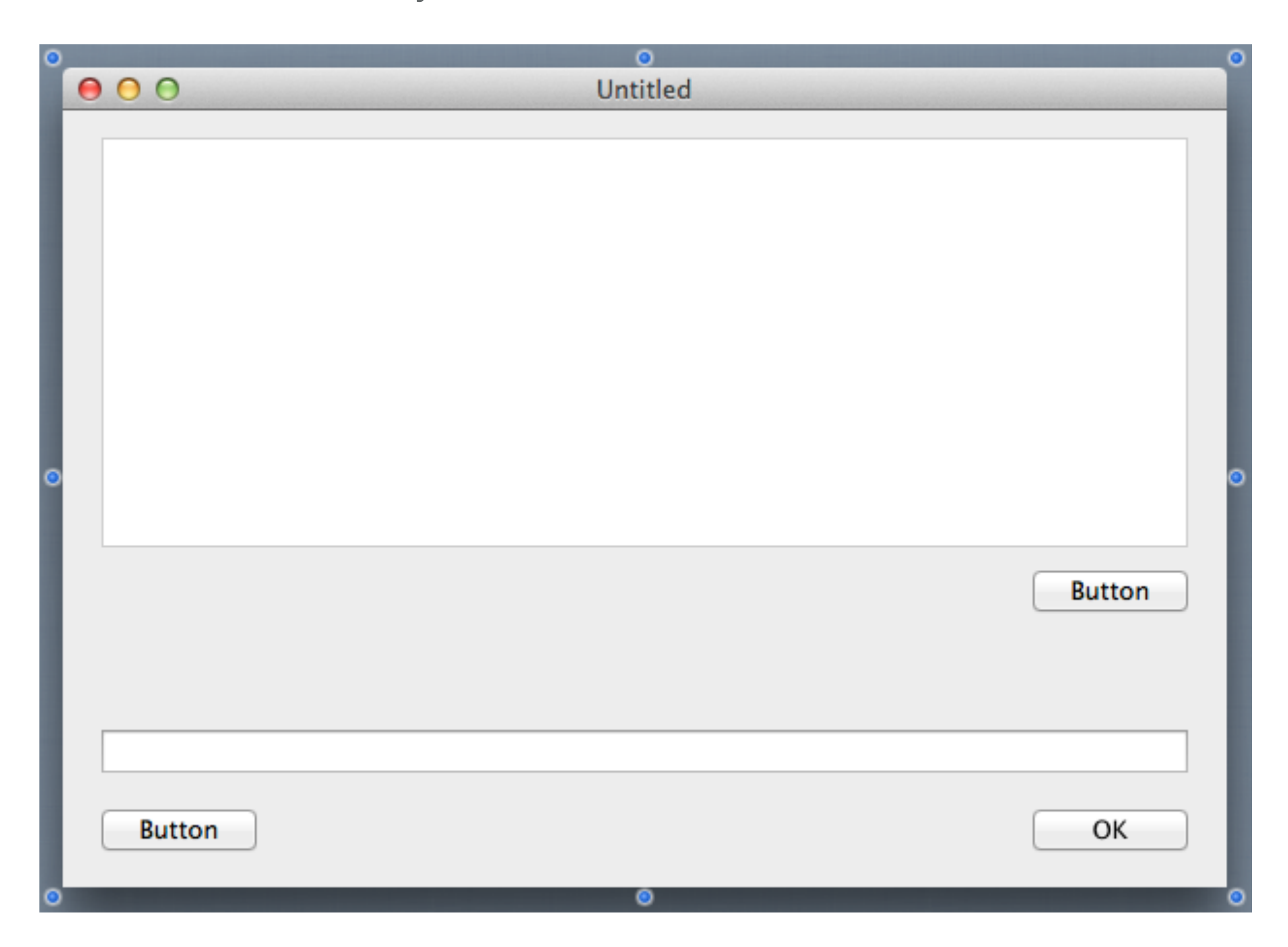
Textfeld (Text Field)

Hinzufügen des URL-Felds

Im URL-Feld kann der Benutzer eine URL eingeben, um diese der Liste hinzuzufügen.

1. Klicken Sie in der Library auf ein Textfeld (Text Field) und ziehen Sie es in das Fenster zwischen Löschen- und Hinzufügen-Knopf.
2. Verändern Sie die Größe des Felds: Ziehen Sie es mittels seiner Anfasser und unterstützt von den intelligenten Hilfslinien auf die Breite der List Box.
3. Ihr Fenster sollen nun ungefähr so aussehen:

Abb. 2.3 Fenster-Layout mit Textfeld



Eigenschaften (Properties)

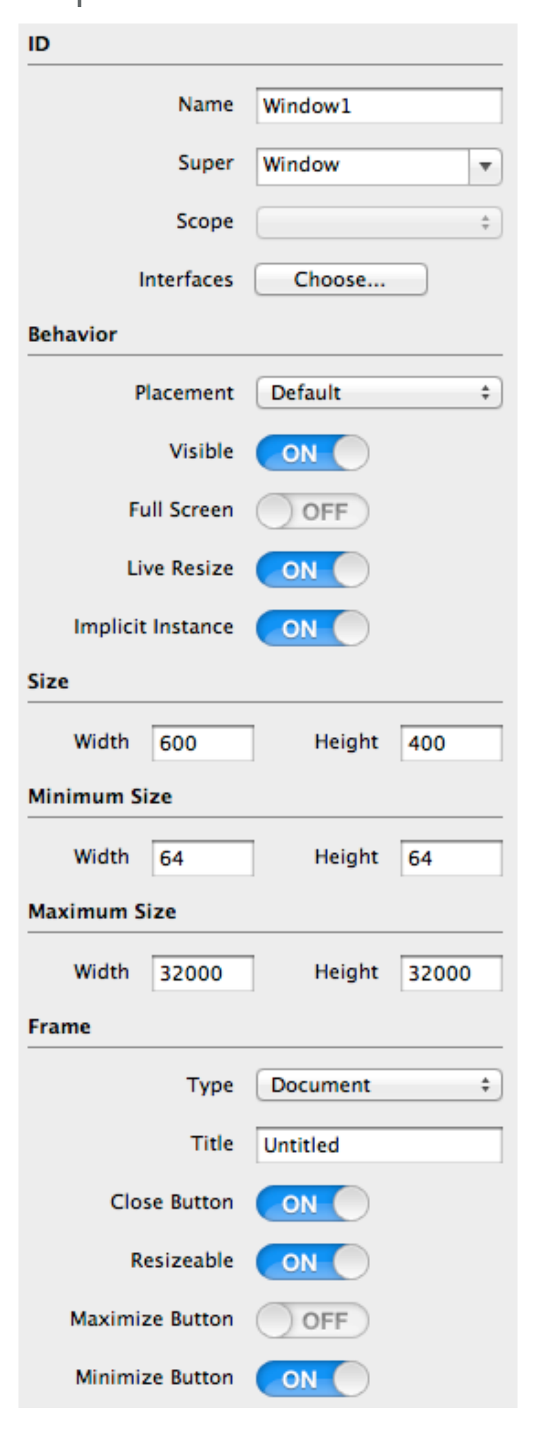
Was ist eine Eigenschaft (Property)?

Eine **Property** ist ein Wert einer Klasse, wie etwa die Breite eines Fensters. Mittels Veränderung der Properties einer **Klasse** können Sie Verhalten und Aussehen einer **Instanz** der Klasse verändern.

In diesem Projekt werden Sie verschiedene Properties des Fensters und seiner Steuerelemente verändern. Dazu gehören:

- Unbenennung aller Steuerelemente (und des Fensters), damit ihre Namen ihre Funktionalität besser beschreiben und sie somit eindeutiger innerhalb des Programmcodes ansprechbar sind.
- Verändern der Schaltflächen-Titel.
- Setzen von Verriegelungs-Properties, um sicherzustellen, dass die Steuerelemente mit Größenveränderung des Fensters korrekt skaliert werden.

Abb. 2.4
Fensterproperties im Inspector



Inspector

Im Inspector werden die Fenster- und Steuerelement-Properties verändert. Der Inspector teilt sich das rechte Panel des Hauptfensters mit der Library. Sie können sich den Inspector anzeigen lassen, indem Sie auf das Inspector-Symbol in der Werkzeugleiste klicken.

Sie können mittels der Tastaturkürzel ⌘-L auf OS X (STRG-K unter Windows und Linux) zur Library und ⌘-I auf OS X (STRG-I unter Windows und Linux) zum Inspector wechseln.

Fenstereigenschaften

Wenn Sie dies noch nicht getan haben sollten, rufen Sie jetzt den Inspector durch Klick auf das Inspector-Symbol in der Werkzeugleiste auf.

Sie sollten nun die Properties **Name** und **Title** ändern:

1. Klicken Sie im Layout-Editor auf die Titelzeile des Fensters, um diese auszuwählen. Der Inspector zeigt jetzt die Properties des Fensters an.

2. Ändern Sie den Text im Feld Name (ganz oben im Inspector in der Gruppe ID) von „Window1“ zu „URLVerwaltungsFenster“. Drücken Sie den *Zeilenschalter* (*Return*), um die Namensänderung im Navigator zu sehen.

3. Ändern Sie den Text im Feld Title (weiter unten in der Frame-Gruppe) von „Untitled“ zu “URL-Verwaltung”.

Abb. 2.5 Namensänderung im Navigator

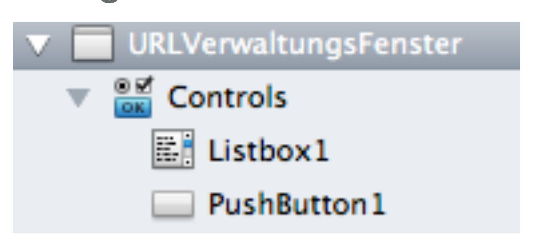
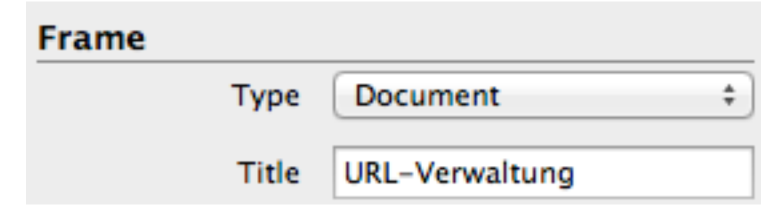
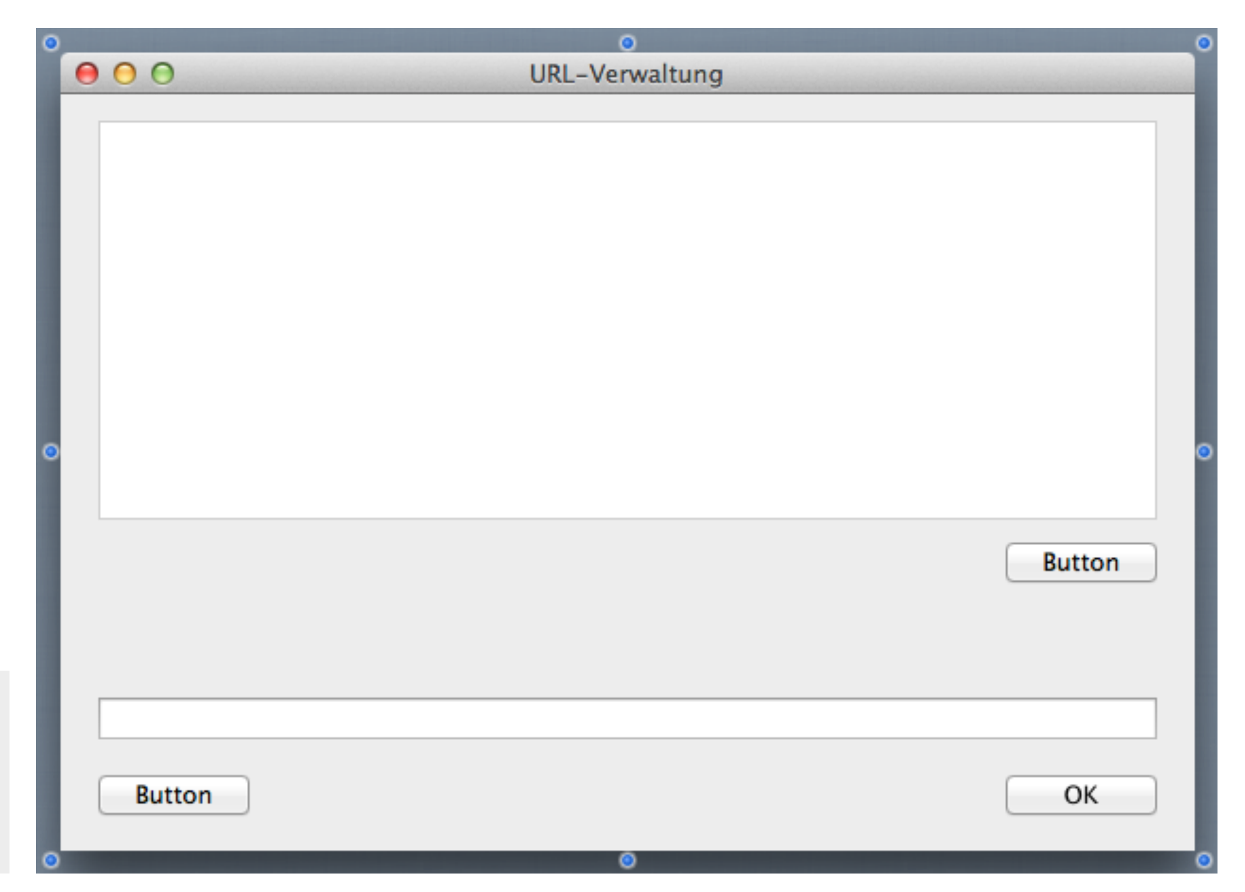


Abb. 2.6 Änderung des Fenstertitels



Nach Eingabe von *Return* sehen Sie den neuen Namen in der Titelzeile des Fensters

Abb. 2.7 Fensterlayout mit neuer Titelzeile



Abschnitt 6

List Box-Eigenschaften

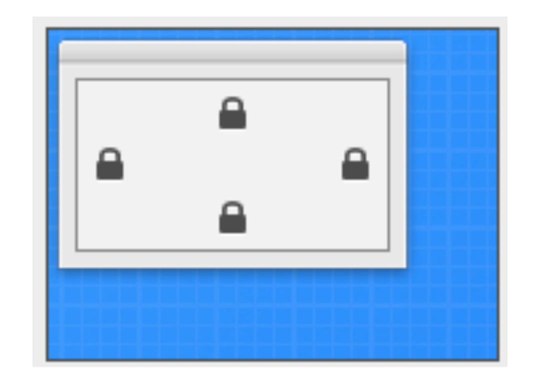
In der List Box sollen die vom Benutzer eingegebenen URLs aufgelistet werden. Es müssen die Properties **Name** und **Locking** geändert werden.

Klicken Sie zunächst im Layout-Editor die Listbox an, um sie auszuwählen. Der Inspector zeigt nun die Properties der Listbox an.

1. Ändern Sie im Feld Name (in der ID Gruppe) den Text von „Listbox1“ in “URLListe”. Drücken Sie *Return*, um die Veränderung in den Navigator zu übernehmen.

2. Jetzt müssen die **Verriegelungspositionen** verändert werden, damit die Listbox bei Größenveränderungen des Fensters mitskaliert wird. Die Locking-Gruppe im Inspector zeigt verriegelte Schloss-Symbole an der oberen und linken Seite des Rahmens, aber geöffnete an der unteren und der rechten Seite.

Abb. 2.8 Verriegelung der Listbox



Klicken Sie auf die Schloss-Symbole, bis alle vier Symbole als verriegelt angezeigt werden.

Schaltflächeneigenschaften

Die drei auf dem Fenster platzierten Buttons sollen Aktionen auslösen. Sie sollten die folgenden Properties jedes Buttons ändern: **Name, Caption** und **Locking.**

Löschen-Schaltfläche

Der Button Löschen wird verwendet, um URLs aus der URLListe zu entfernen.

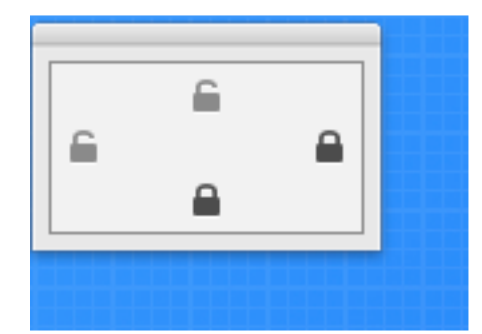
1. Klicken Sie im Layout-Editor zunächst auf den Löschen-Button, um ihn auszuwählen (das ist der Button direkt unterhalb der Listbox). Der Inspector zeigt nun die Properties des Löschen-Buttons an.

2. Ändern Sie im Feld Name des Inspectors (in der ID-Gruppe) den Text von „PushButton1" in „LöschButton". Drücken Sie *Return*, um die Veränderung in den Navigator zu übernehmen.

3. Geben Sie im Feld **Caption** (= Titel, Beschriftung) (unten im Inspector in der Gruppe Appearance (Erscheinungsbild)) statt „Button" „Löschen" ein. Drücken Sie *Return*, um die Eingabe zu übernehmen und die Veränderung im Button des Fensters zu sehen.

4. Nun sollten Sie das Locking ändern, damit der Löschen-Button bei Größenveränderungen des Fensters stets auf der rechten Seite unterhalb der Listbox bleibt.

Abb. 2.9 Locking für den LöschButton



Die Locking-Gruppe im Inspector zeigt verriegelte Schloss-Symbole an der oberen und linken Seite des Rahmens und geöffnete an der unteren und der rechten Seite. Klicken Sie auf die Symbole, bis oberes und linkes Symbol geöffnet sowie rechtes und unteres Symbol verriegelt sind.

Hinzufügen-Schaltfläche

Mithilfe des Hinzufügen-Buttons wird die im Text Field eingegebene URL der URLListe hinzugefügt.

1. Klicken Sie im Layout-Editor auf den Hinzufügen-Button, um ihn auszuwählen (dies ist die Schaltfläche in der linken unteren Ecke des Fensters unterhalb des TextFields). Der Inspector zeigt nun die Properties des Pushbuttons an.

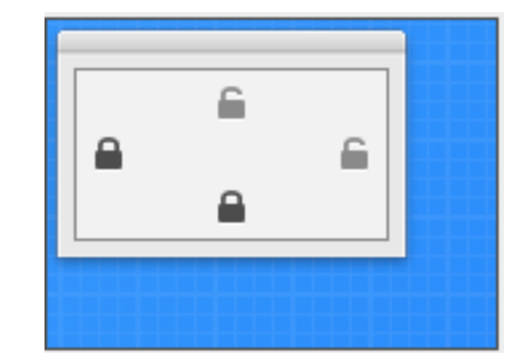
2. Ändern Sie im Feld Name des Inspectors (in der ID-Gruppe) den Text von „PushButton2“ in „HinzufügenButton“. Drücken Sie *Return*, um die Eingabe zu übernehmen und die Veränderung im Navigator zu sehen.

3. Geben Sie im Feld Caption (unten im Inspector in der Gruppe Appearance) statt „Button“ „Hinzufügen“ ein. Drücken Sie *Return*, um die Eingabe zu übernehmen und die Veränderung im Button des Fensters zu sehen. Ziehen Sie den Button im Layout-Editor ggf. ein bisschen breiter, um den ganzen Text anzuzeigen.

4. Jetzt sollten Sie die Verriegelungspositionen des Buttons überprüfen, damit die Schaltfläche bei Größenveränderungen des Fensters immer in der linken unteren Ecke bleibt.

   Die Locking-Gruppe im Inspector zeigt verriegelte Schloss-Symbole an der oberen und linken Seite des Rahmens und geöffnete an der unteren und der rechten Seite.
   Klicken Sie auf die Symbole, bis oberes und rechtes Symbol geöffnet sowie linkes und unteres Symbol verriegelt sind.

Abb. 2.10 Locking für den HinzufügenButton



Anzeigen-Schaltfläche

Ein Klick auf den Button Anzeigen soll den Webbrowser öffnen und die im TextField eingegebene URL anzeigen.

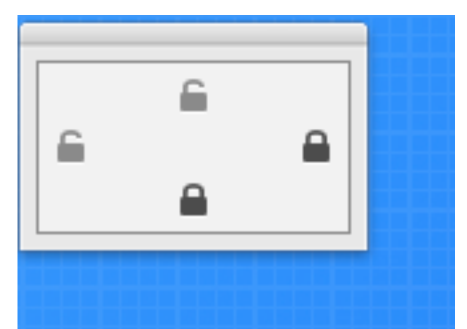
1. Klicken Sie im Layout-Editor auf den Anzeigen-Button, um ihn auszuwählen (dies ist die Schaltfläche in der rechten unteren Ecke des Fensters unterhalb des TextFields). Der Inspector zeigt nun die Properties des Pushbuttons an.

2. Ändern Sie im Feld Name des Inspectors (in der ID-Gruppe) den Text von „PushButton3“ in „AnzeigeButton“. Drücken Sie *Return*, um die Eingabe zu übernehmen und die Veränderung im Navigator zu sehen.

3. Geben Sie im Feld Caption (unten im Inspector in der Gruppe Appearance) statt „Button“ „Anzeigen“ ein. Drücken Sie *Return*, um die Eingabe zu übernehmen und die Veränderung im Button des Fensters zu sehen.

4. Jetzt sollten Sie die Verriegelungspositionen des Buttons überprüfen, damit die Schaltfläche bei Größenveränderungen des Fensters immer in der rechten unteren Ecke bleibt.

Abb. 2.11 Locking für den AnzeigeButton



Die Locking-Gruppe im Inspector zeigt verriegelte Schloss-Symbole an der oberen und linken Seite des Rahmens und geöffnete an der unteren und der rechten Seite.
Klicken Sie auf die Symbole, bis oberes und linkes Symbol geöffnet sowie rechtes und unteres Symbol verriegelt sind.

Text Field-Eigenschaften

Im Text Field wird der Anwender die URLS angeben, die der Liste hinzugefügt werden sollen. Hier werden auch die URLs angezeigt, die in der Liste angeklickt werden.
Die Properties **Name** und **Locking** müssen modifiziert werden.

1. Klicken Sie im Layout-Editor zunächst auf das Text Field, um es auszuwählen. Der Inspector zeigt nun die Text Field-Properties an.

2. Ändern Sie im Feld Name des Inspectors (in der ID-Gruppe) den Text von „TextField1“ in „URLField“. Drücken Sie *Return*, um die Veränderung in den Navigator zu übernehmen.

3. Nun sollten Sie das Locking ändern, damit das Text Field bei Größenveränderungen des Fensters mitskaliert wird.
   Die Locking-Gruppe im Inspector zeigt verriegelte Schloss-Symbole an der oberen und linken Seite des Rahmens und geöffnete an der unteren und der rechten Seite. Klicken Sie auf die Symbole, bis das obere Symbol geöffnet ist und das linke, rechte und untere Symbol verriegelt sind.

Abb. 2.12 Locking für das URLField



Projekttest

Das Projekt sichern

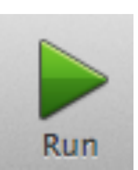
Sie sollten Ihr Projekt regelmäßig speichern – und grundsätzlich vor dem Ausführen.

1. Speichern Sie das Projekt via File ➝ Save.

2. Benennen Sie das Projekt „DesktopTutorial“ und klicken Sie auf Save.

Das Projekt ausführen

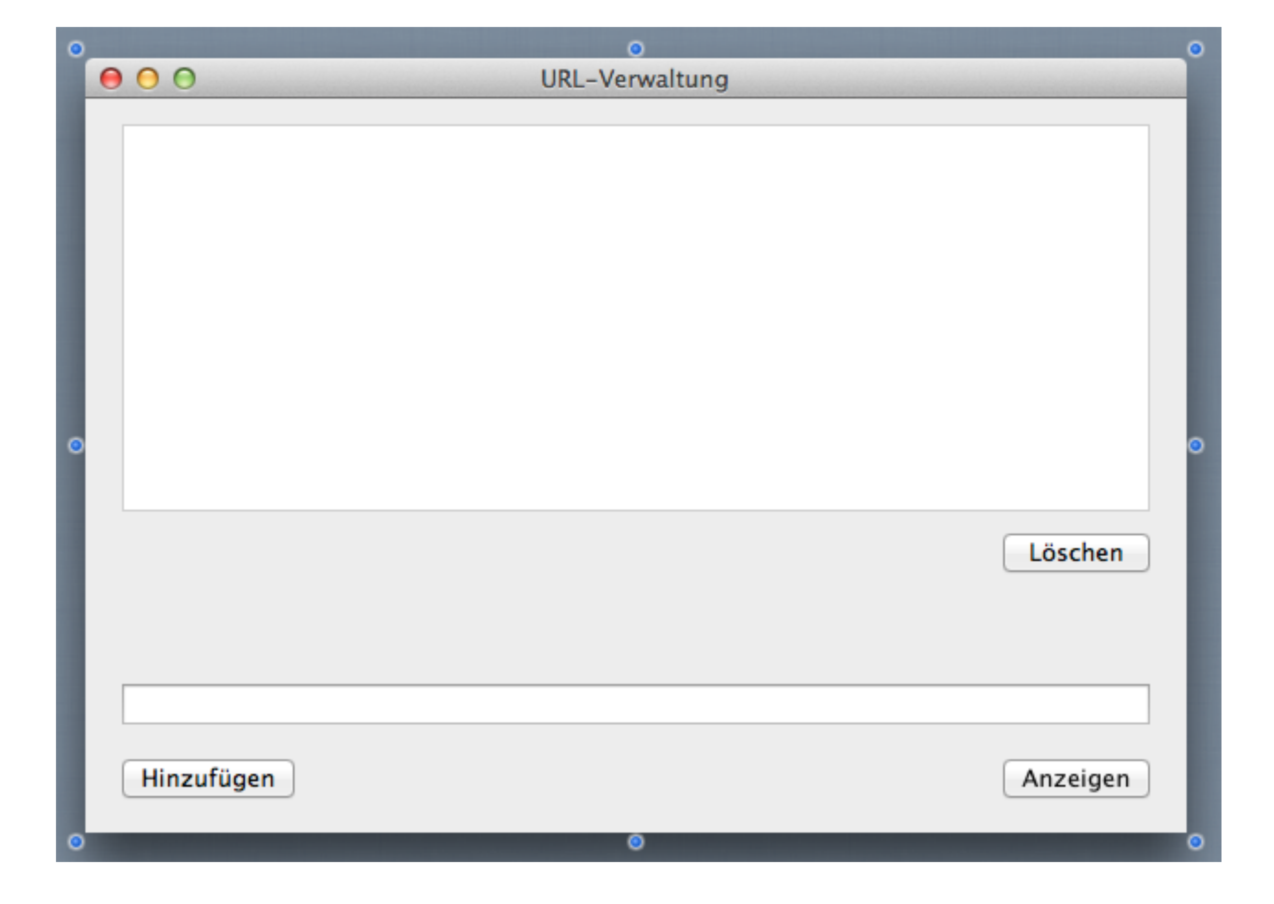
Jetzt, da Ihr Benutzerinterface-Layout vollständig ist, können Sie Ihr Programm testen. Klicken Sie auf das Run-Symbol in der Werkzeugleiste, um es ausführen zu lassen.



Sobald das URL-Verwaltungsfenster angezeigt wird, können Sie mit den Schaltflächen interagieren, indem Sie auf sie klicken. Sie können Text in das Text Field eingeben und das Fenster vergrößern und verkleinern und beobachten, wie sich die Locking-Einstellungen dabei auf die Elemente auswirken.

Allerdings verfügt die URL-Verwaltung über keinerlei Funktionalität. Wir müssen Programmcode hinzufügen, worum es im nächsten Kapitel gehen wird.

Abb. 2.13 Das vollständige URL-Verwaltungsfenster-Layout



Kapitel 3

Code hinzufügen

Der abschließende Schritt in der Erstellung der URL-Verwaltung: Das Hinzufügen von Programmcode.



Anzeigen-Schaltfläche

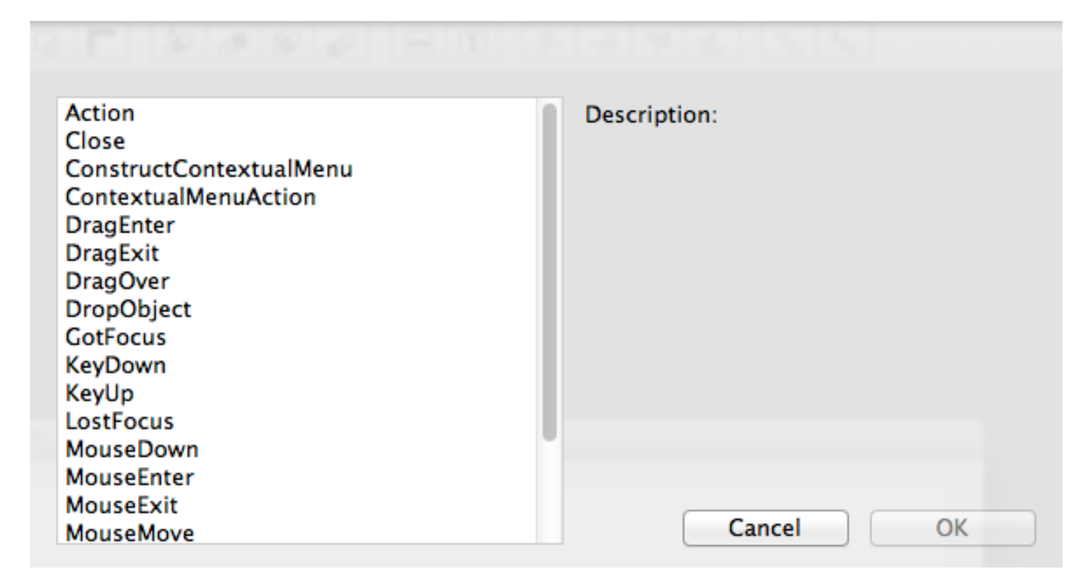
Code zum AnzeigenButton hinzufügen

Das erste Steuerelement, das wir mit Code ergänzen, wird der AnzeigeButton sein. Ein Klick auf diese Schaltfläche soll den voreingestellten Webbrowser öffnen und die in das URLField eingegebene URL anzeigen:

Ergänzen Sie den Code folgendermaßen:

1. Doppelklicken Sie im Layout-Editorfenster das **AnzeigeButton-**Steuerelement, das Sie „Anzeigen" betitelt haben.
   Der **Event-Handler-Hinzufügen-Dialog** (**Add Event Handler**) erscheint.

   Abb. 3.1 Event-Handler-Hinzufügen-Dialog

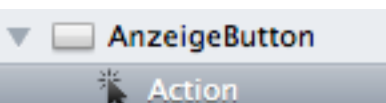
   

   Wenn ein Benutzer auf einen PushButton klickt, wird der Programmcode im **Action-Event**-Handler ausgeführt.

2. Sie wollen eine Aktion programmieren, die bei Klicken des Buttons ausgeführt wird. Wählen Sie daher Action aus der Event-Handler-Liste und klicken Sie auf OK.

3. Wie Sie sehen, zeigt der Navigator den Action-Event nun unterhalb des AnzeigeButton-Steuerelements an, und der **Code-Editor** erscheint.

   Abb. 3.2 Action-Event-Handler für den AnzeigeButton

4. Der Programmierbefehl, um eine URL im voreingestellten Webbrowser des Benutzers zu öffnen, lautet **ShowURL**. Um den Knopf zu testen, sorgen wir zunächst dafür, dass bei einem Klick auf den Button die stets gleiche URL geöffnet wird:

```
ShowURL("http://www.xojo.com")
```

ShowURL ist eine **globale Methode**, da sie keinem besonderen Objekt zugeordnet ist. Sie kann von überall in der Anwendung aufgerufen werden.

5. Starten Sie die Anwendung und klicken Sie auf den AnzeigeButton. Ihr voreingestellter Webbrowser sollte nun die Xojo-Homepage anzeigen.
   Dies ist als Funktion natürlich nicht sonderlich sinnvoll. Wir müssen den Programmcode modifizieren, damit er die URL verwendet, die in das URLField eingegeben wurde.

6. Beenden Sie die Anwendung, um in das Editor-Fenster zurückzugelangen (Wählen Sie Datei ↠ Beenden auf Windows oder Linux bzw. URL-Verwaltung.debug ↠ Beenden auf OS X).

7. Kehren Sie im Code-Editor in den ShowButton-Action-Event zurück und löschen Sie den bisher eingegebenen Code.

   Sie müssen also die URL erhalten, die ins URLField eingegeben wurde.
   Eventuell denken Sie gerade, dass Sie den Text erhalten können, indem Sie den Namen des Felds, URLField, adressieren.
   Das ist nah dran, trifft es aber nicht völlig.

   Stattdessen müssen Sie eine Property von URLField ansprechen.

   Wann immer Sie auf eine Property eines Objekts zugreifen wollen, verwenden Sie den Namen des Objekts, gefolgt von einem Punkt und dahinter den Namen der Eigenschaft.
   Kurz gesagt: Verwenden Sie diese Syntax:

   ObjektName.PropertyName

   Dies nennt man **Punktnotation** (**dot notation**), eine häufig benutzte Schreibweise in objektorientierter Programmierung.

   In diesem Fall heißt das Objekt **URLField** und die gesuchte Property heißt **Text** (schauen Sie in die Sprachreferenz, um alle Properties eines Tetxtfields kennenzulernen).

   Jetzt können Sie den Code

```
ShowURL(URLField.Text)
```

   eingeben.
   (Sollte sich dies als schwierig erweisen, stellen Sie bitte sicher, dass Sie die UL-Verwaltung wie in Schritt 5 angegeben wirklich beendet haben.)

8. Speichern Sie das Projekt durch Wahl von File ↠ Save.

9. Starten Sie die App und geben Sie eine beliebige andere URL in das URLField ein, wie z. B. „http://www.xojoblog.com“.

10. Klicken Sie anschließend auf den AnzeigeButton
Ihr voreingestellter Webbrowser sollte nun die eingegebene Website öffnen.

11. Beenden Sie die Anwendung, um in den Editor zurückzugelangen (Wählen Sie Datei ↠ Beenden auf Windows oder Linux bzw. URL-Verwaltung.debug ↠ Beenden auf OS X).

Hinzufügen-Schaltfläche

Code zum HinzufügenButton hinzufügen

Mittels des HinzufügenButton werden URLs der Liste hinzugefügt. Der Code, den Sie der Schaltfläche hinzufügen müssen, muss also entgegennehmen, was in das URLField eingetippt wurde und dieses in einer neuen Zeile an die Liste anfügen.

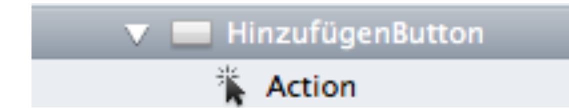
Gehen Sie dazu wie folgt vor:

1. Doppelklicken Sie im Fenster des Layout-Editors das **HinzufügenButton**-Steuerelement, das Sie „Hinzufügen" genannt haben.

   Der **Event-Handler-Hinzufügen-Dialog** (**Add Event Handler**) erscheint. Wie bereits beim AnzeigenButton wollen Sie den **Action**-Event-Handler verwenden, um Ihren Code ausführen zu lassen, wenn der Benutzer auf die Schaltfläche klickt. Wählen Sie also Action aus der Event-Handler-Liste und klicken Sie auf OK.

   Im Navigator erscheint der Action-Event unterhalb des HinzufügenButton-Steuerelements und der Code-Editor zeigt den (noch leeren) Code des Events an.

Abb. 3.3 Action-Event-Handler für den HinzufügenButton



2. Eine Zeile wird einer Listbox mit der **AddRow**-Methode hinzugefügt.
   Sie wissen bereits, wie Sie an den Text eines TextFields gelangen. Kombinieren Sie dies mit AddRow und Sie erhalten diesen Code:

```
URLListe.AddRow(URLField.Text)
```

   Wie vorher schon erwähnt, können Objekte Properties besitzen. Nun sehen Sie, dass Objekte ebenfalls Methoden besitzen können. AddRow ist eine der vielen Methoden, die einem Listbox-Steuerelement zuer Verfügung stehen.

3. Speichern Sie das Projekt durch Wahl von File → Save.

4. Starten sie die App, um sie zu testen. Geben Sie URLs in das URLField ein und klicken Sie auf den HinzufügenButton. Sie tauchen daraufhin in der URLListe auf.

5. Beenden Sie die Anwendung, um in den Editor zurückzugelangen (Wählen Sie Datei ↠ Beenden auf Windows oder Linux bzw. URL-Verwaltung.debug ↠ Beenden auf OS X).

Löschen-Schaltfläche

Code dem LöschButton hinzufügen

Mit dem LöschButton werden URLs aus der Liste entfernt. Der von Ihnen zu schreibende Code muss die ausgewählte Zeile der Liste erkennen und diese aus ihr löschen können.

Gehen Sie dazu wie folgt vor:

1. Doppelklicken Sie im Fenster des Layout-Editors den **Lösch-Button**, den Sie „Löschen" betitelt haben..

   Der **Event-Handler-Hinzufügen-Dialog** (**Add Event Handler**) erscheint. Wie bereits bei den anderen Schaltflächen wollen Sie den **Action**-Event-Handler verwenden, um Ihren Code ausführen zu lassen, wenn der Benutzer auf die Schaltfläche klickt. Wählen Sie also Action aus der Event-Handler-Liste und klicken Sie auf OK.

   Im Navigator erscheint der Action-Event unterhalb des Lösch-Button-Steuerelements und der Code-Editor zeigt den (noch leeren) Code des Events an.

2. Ob eine Zeile aus einer Listbox zu entfernen, müssen Sie zunächst bestimmen, ob und wenn ja, welche Zeile in ihr ausgewählt ist.
   In einer Listbox findet sich die aktuell ausgewählte Zeile in der **ListIndex**-Property.

3. Verwenden Sie die Listbox-Methode **RemoveRow** , um eine Zeile aus einer Listbox zu entfernen. Sie übergeben RemoveRow die Zeilennummer der zu entfernenden Zeile als Parameter.
   Ihr Code sieht nun also so aus:

```
URLListe.RemoveRow(URLListe.ListIndex)
```

4. Speichern Sie das Projekt durch Wahl von File → Save.

5. Starten sie die App, um sie zu testen. Geben Sie URLs in das URLField ein und klicken Sie auf den HinzufügenButton, um sie in die Liste einzufügen.
   Klicken Sie nun auf eine URL in der Liste und dann auf den LöschButton. Die Zeile verschwindet.

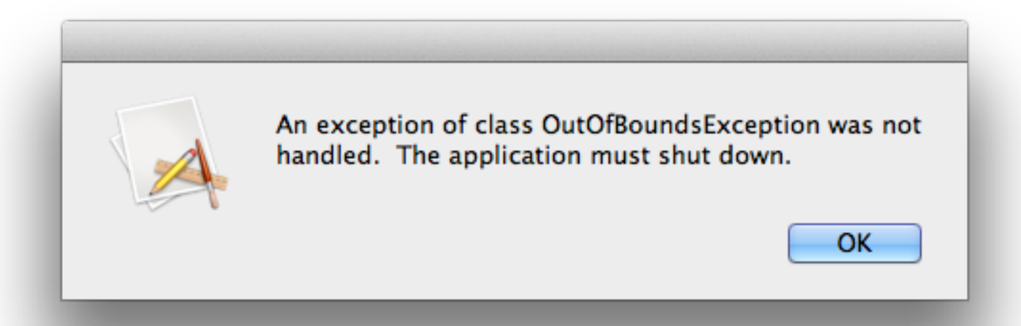
Aber es gibt einen Bug in Ihrer App.
Was passiert, wenn der Anwender auf den LöschButton klickt, ohne eine Zeile in der Liste anzuwählen?
Versuchen Sie es! Ihre App wird in den Editor zurückschalten und eine Zeile optisch hervorheben.

Abb. 3.4 Ein Laufzeitfehler im Debugger

```
Sub Action()
  URLListe.RemoveRow ( URLListe.ListIndex )
End Sub
```

Ihr Programm ist durch eine **OutOfBounds-Exception** abgestürzt und Sie befinden sich jetzt im **Debugger**.
Der Fehler trat auf, weil Sie versuchten, eine Zeile zu entfernen, die nicht existiert. Wenn keine Zeile in einer Listbox ausgewählt ist, liefert die Listindex-Property den Wert -1 zurück.
Wenn Sie auf das Resume-(Fortsetzen-)Symbol in der Werkzeugleiste klicken, erscheint die Fehlermeldung der Anwendung.

Abb. 3.5 Laufzeit-Fehlermeldung



6. Da die Exception nicht von einer programmeigenen Fehlerbehandlungsmethode abgefangen wurde und der interne Programmzustand nunmehr unsicher ist, beendet sich das Programm, wenn Sie die Fehlermeldung bestätigen, und Sie kehren in den Editor zurück.

7. Nun können Sie Ihren Code ergänzen, um den Fehler zu vermeiden. Sie wollen die RemoveRow-Methode nicht aufrufen, wenn keine Zeile angewählt ist.
   Der Programmcode dafür sieht folgendermaßen aus:

   ```
   If URLListe.ListIndex >= 0 Then
     URLListe.RemoveRow(URLListe.ListIndex)
   End If
   ```

   Hiermit stellt das Programm durch Überprüfung der ListIndex-Property sicher, dass eine Zeile angewählt ist, um nur dann diese Zeile zu löschen.

8. Speichern Sie das Projekt durch Wahl von File → Save.

9. Starten Sie das Projekt erneut und klicken Sie auf den Lösch-Button, ohne zuvor eine Zeile in der URLListe angewählt zu haben. Kein Absturz mehr! Sie haben den Fehler **debuggt**.

Die Listbox

Code der Listbox hinzufügen

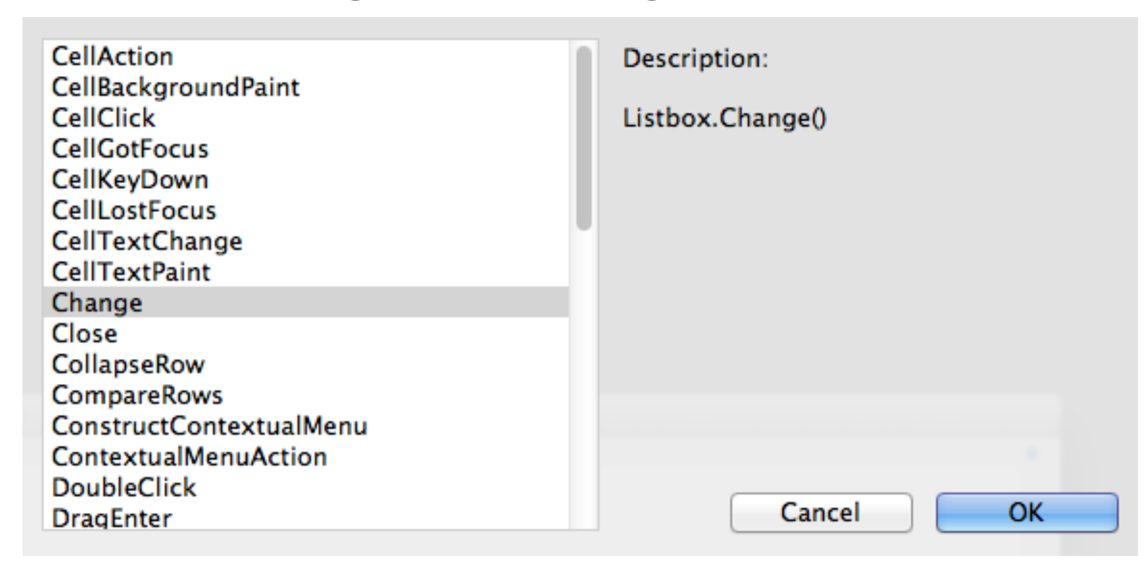
Das letzte Steuerelement, das noch Code benötigt, ist die URLListe. Alle hinzugefügten URLs werden hier angezeigt. Klickt der Benutzer auf eine URL in der Liste, sollte diese in das URLField übertragen werden.

Gehen Sie wie folgt vor:

1. Doppelklicken Sie im Fenster des Layout-Editors das **URLList**-Steuerelement.

   Der **Event-Handler-Hinzufügen-Dialog** (**Add Event Handler**) erscheint, diesmal mit einer anderen Liste als der, die Sie von den PushButtons her kennen.
   Sie wollen die URL in das Text Field übertragen, wenn der Benutzer auf eine Zeile klickt.
   Der **Change**-Event-Handler wird aufgerufen, wann immer der Anwender eine neue Zeile auswählt.
   Wählen Sie also Change aus der Event-Handler-Liste und klicken Sie auf OK.
   Im Navigator erscheint der Change-Event unterhalb des

Abb. 3.6 Hinzufügen eines Change-Event Handlers



   URLListe-Steuerelements und der Code-Editor zeigt den (noch leeren) Code des Events an.

2. Der Text einer ausgewählten Listbox-Zeile befindet sich in der **Text**-Property der Listbox.
   Sie wissen bereits, dass die Textfield.Text-Property den Text beinhaltet, den der Benutzer in das Textfield eingegeben hat, ebenso wie Sie dieser Property einen Wert zuweisen können, um das Feld via Programmcode mit Text zu füllen.

Ihr Code sieht also wie folgt aus:

```
URLField.Text = Me.Text
```

Beachten Sie bitte, dass wir hier **Me.**Text verwenden, um den Text der ausgewählten Zeile der Listbox zu erhalten. Wir hätten die Eigenschaft ebenso über URLListe.Text ansprechen können. Warum also Me?

Da Sie sich hier im **Event**-Handler der Listbox befinden, können Sie alle Methoden und Eigenschaften der Listbox über Me aufrufen. Damit können Sie z.B. ein Listbox-Steuerelement umbenennen, ohne dass Sie seinen Code überarbeiten und dort den neuen Namen des Elements eingeben müssen.

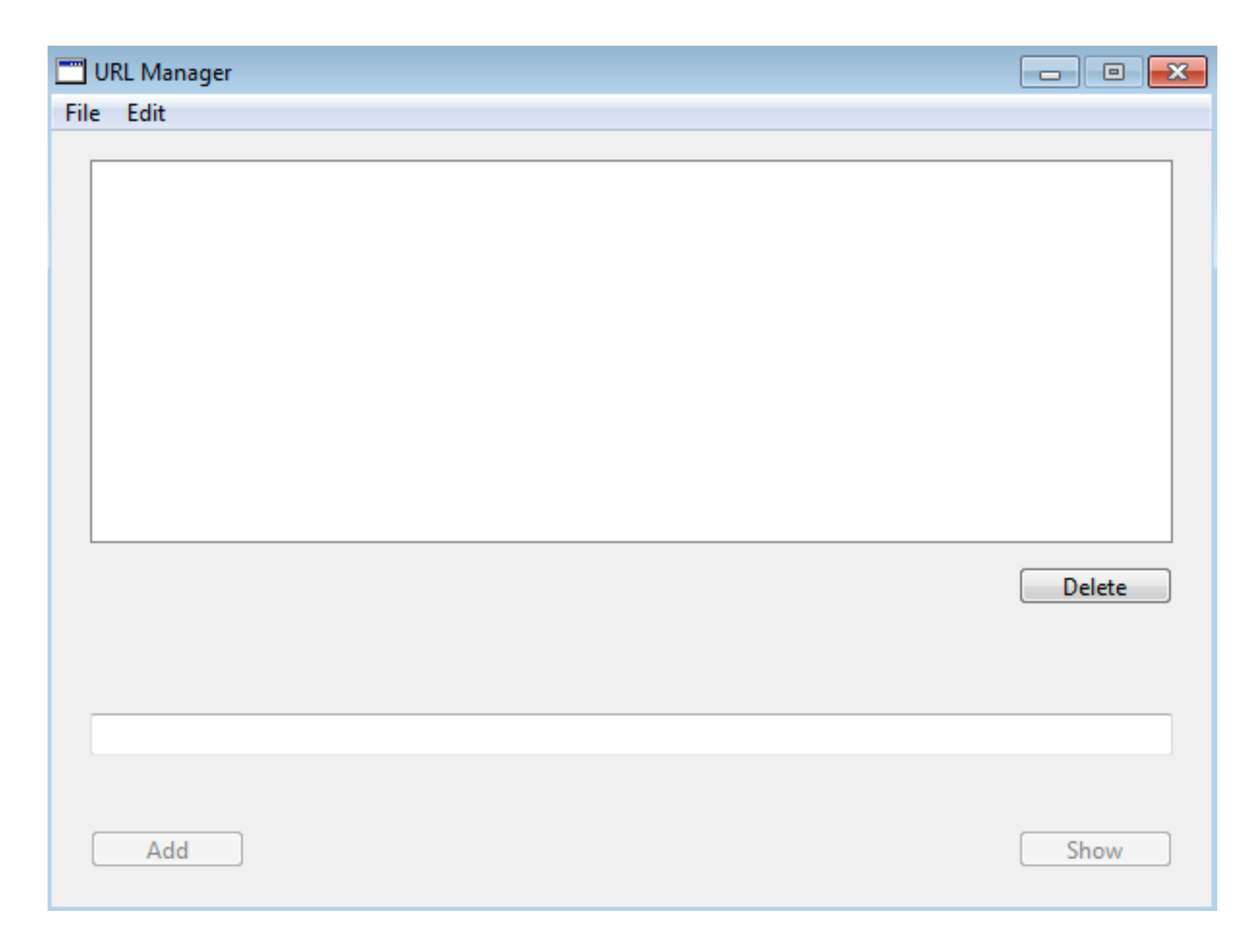
3. Speichern Sie das Projekt durch Wahl von File ↠ Save.

4. Starten sie die App, um sie zu testen. Geben Sie einige URLs in das URLField ein und klicken Sie auf den HinzufügenButton, um sie in die Liste einzufügen. Klicken Sie dann auf eine bereits eingefügte URL und bemerken sie, dass diese nun im URLField angezeigt wird.

Beenden Sie die Anwendung, um in den Editor zurückzugelangen (Wählen Sie Datei ↠ Beenden auf Windows oder Linux bzw. URL-Verwaltung.debug ↠ Beenden auf OS X).

Das war's! Ihre Anwendung ist jetzt fertig!

Abb. 3.7 URL-Verwaltung unter Windows (englische Version)



Kapitel 4

Die nächsten Schritte

Da Sie nun eine fertige Anwendung erstellt haben, ist es an der Zeit, sie zu testen und vielleicht ein paar Verbesserungen einzuführen.



Die URL-Verwaltung testen

Unabdingbar: Der Programmtest

Sie sind nicht wirklich fertig mit dem Programmieren, nur weil Sie nun eine fertige Applikation erstellt haben. Ein guter Entwickler testet seine Anwendungen stets gründlich, um mögliche Probleme aufzuspüren.

Ein Problem haben Sie bereits gefunden und behoben – den Klick auf den Löschbutton ohne ausgewählte Listenzeile. Können Sie sich vorstellen, dass noch weitere Probleme existieren, die korrigiert werden sollten?

Starten Sie ihre App und spielen Sie ein wenig damit herum. Notieren Sie sich Dinge, die Sie gerne ändern möchten.

Im nächsten Abschnitt werden wir die URL-Verwaltung ein wenig verbessern.

Abschnitt 2

Verbesserungen

Schaltflächenbedienbarkeit

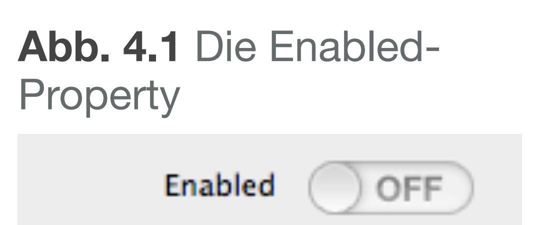
Ist ihnen aufgefallen, dass einige Buttons in der URL-Verwaltung manchmal nicht verfügbar sein sollten?
Der AnzeigeButton beispielsweise sollte nicht versuchen, seine Action-Methode auszuführen, wenn keine URL ins URLField eingetragen wurde. Und Sie wollen sicher keine URL in die Liste übernehmen, wenn das URLField leer ist.

Es gibt verschiedene Methoden, dies zu erreichen. Eine davon besteht darin, die Schaltflächen zu deaktivieren, wenn sie nicht benutzbar sein sollen.

Gehen Sie dazu wie folgt vor:

1. Wählen Sie im Layout-Editor den AnzeigeButton, den Sie „Anzeigen" benannt haben. Stellen Sie im Inspector (in der Appearance-Gruppe) die Property Enabled auf Aus (Off).

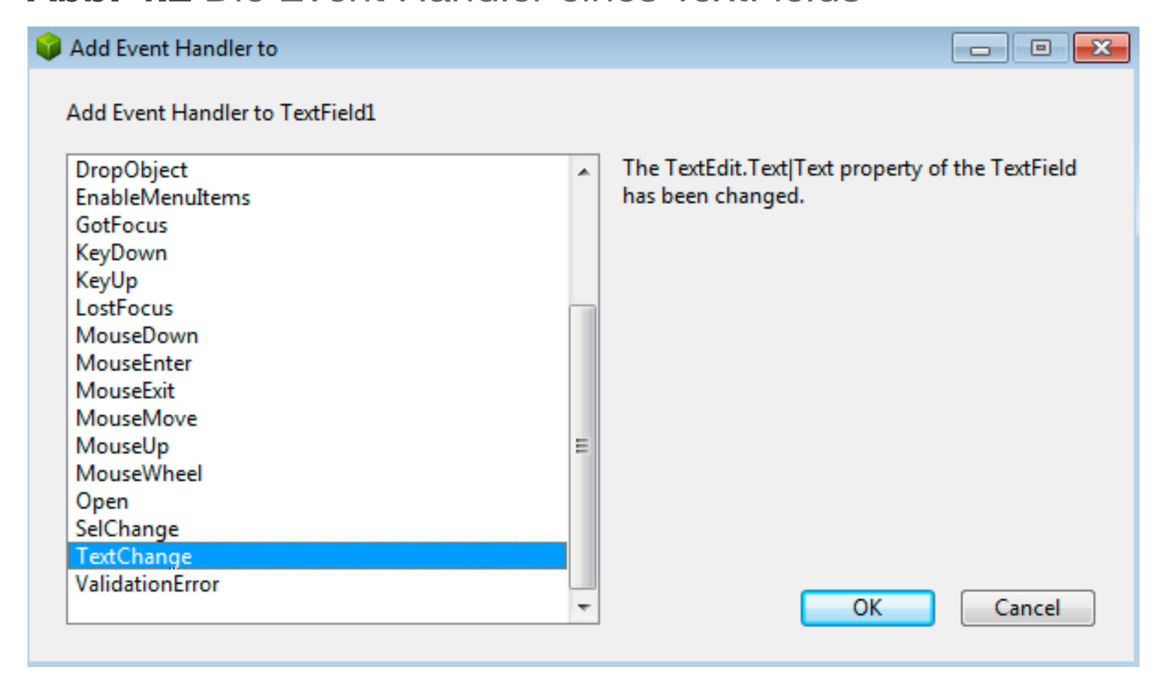
Abb. 4.1 Die Enabled-Property



2. Gehen Sie genauso mit dem HinzufügenButton um.

3. Jetzt werden Sie Programmcode hinzufügen, um die Schaltflächen zu aktivieren, wenn sich Text im URLField befindet.
   Doppelklicken Sie im Fenster auf das **URLField**-Steuerelement. Der **Event-Handler-Hinzufügen-Dialog** (**Add Event Handler**) erscheint, wiederum mit einer anderen Liste als den bisher gesehenen. Jedes Steuerelement besitzt seine individuelle Event-Liste.

Abb. 4.2 Die Event Handler eines TextFields



Wir wollen HinzufügenButton und AnzeigeButton deaktivieren, wenn kein Text im URLField steht.

Der **TextChange**-Event wird bei jeder Veränderung der Text-Property eines Textfields aufgerufen, sei es durch Benutzereingabe oder via Programmcode.
Wählen Sie also TextChange aus der Event-Handler-Liste und klicken Sie auf OK..

Im Navigator erscheint der TextChange-Event unterhalb des URLListe-Steuerelements und der Code-Editor zeigt den (noch leeren) Code des Events an.

4. Geben Sie z.B. diesen Code ein:

```
If Me.Text <> "" Then
  AnzeigeButton.Enabled = True
  HinzufügenButton.Enabled = True
Else
  AnzeigeButton.Enabled = False
  HinzufügenButton.Enabled = False
End If
```

   Dieser Code überprüft die Text-Eigenschaft des Textfields (Me.Text), um zu sehen, ob irgendetwas darin steht. Falls dem so ist, werden AnzeigeButton und HinzufügenButton aktiviert, indem ihre **Enabled**-Properties auf Wahr (**True**) gesetzt werden. Andernfalls werden die Enabled-Properties auf Falsch (**False**) gesetzt.

5. Speichern Sie das Projekt durch Wahl von File ↠ Save.

6. Starten sie die App, um sie zu testen. Nun sind HinzufügenButton und AnzeigeButton beim Programmstart deaktiviert. Sobald Sie Text in das URLField eingeben, werden sie aktiviert. Und entfernen Sie den Texteintrag, geschieht das Umgekehrte.

Abschnitt 3

Erzeugen einer Stand-Alone-App

Weitergabe Ihrer Anwendung

Da Sie nun eine prima Anwendung erstellt haben, möchten Sie sie wahrscheinlich in die Welt entlassen. Dazu müssen Sie eine **Stand-Alone-App** erzeugen.

Mit Xojo können Sie Desktop-Anwendungen für Mac OS X, Windows und Linux erzeugen. Zunächst also sollten Sie entscheiden, für welche Plattform Sie den **Build** erstellen wollen. Diese Auswahl treffen Sie im BUILD-Abschnitt unten im Navigator

Setzen Sie ein Häkchen vor jede Plattform, für die Sie einen Build erstellen wollen. „Dieser Computer („This Computer“) ist vorausgewählt, um Ihnen zu vereinfachen, schnell einen Build für die aktuelle Plattform zu erstellen.

Abb. 4.3 Build Einstellungen

BUILD SETTINGS
Shared
OS X
Windows
Linux
This Computer

In den „Gemeinsamen Einstellungen“ („Shared Settings“) finden Sie plattformübergreifende Parameter. Plattformspezifische Einstellungen können Sie durch Klick auf den Namen einer Plattform vornehmen.

Gemeinsame Einstellungen (Shared Settings)

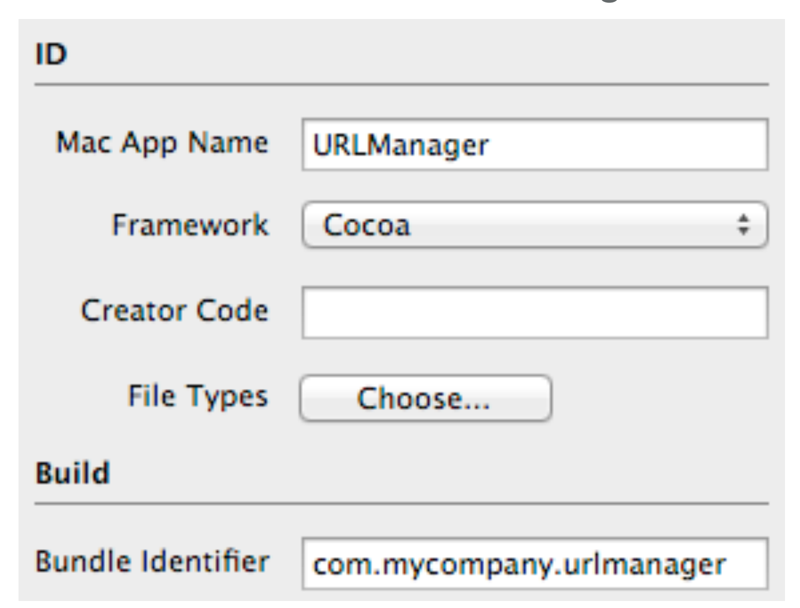
Die gemeinsamen Build-Einstellungen (**Shared Build Settings**) beinhalten die Versionsinformationen, Einstellungen für das Build-Verzeichnis und die Möglichkeit, zu Debugzwecken die Funktionsnamen in den Build zu integrieren.

Anm. des Übersetzers: Die Bedeutungen der Parameter der folgenden Build-Einstellungen finden Sie im Xojo-Benutzerhandbuch und in den Programmierhandbüchern/Online-Bibliotheken des jeweiligen Betriebssystems.

OS X

In den OS X-Build-Einstellungen können Sie den Namen der Mac-Applikation angeben, die Architektur (Carbon oder Cocoa),den Creator Code, Dateitypen und Bundle-Identifizierer.

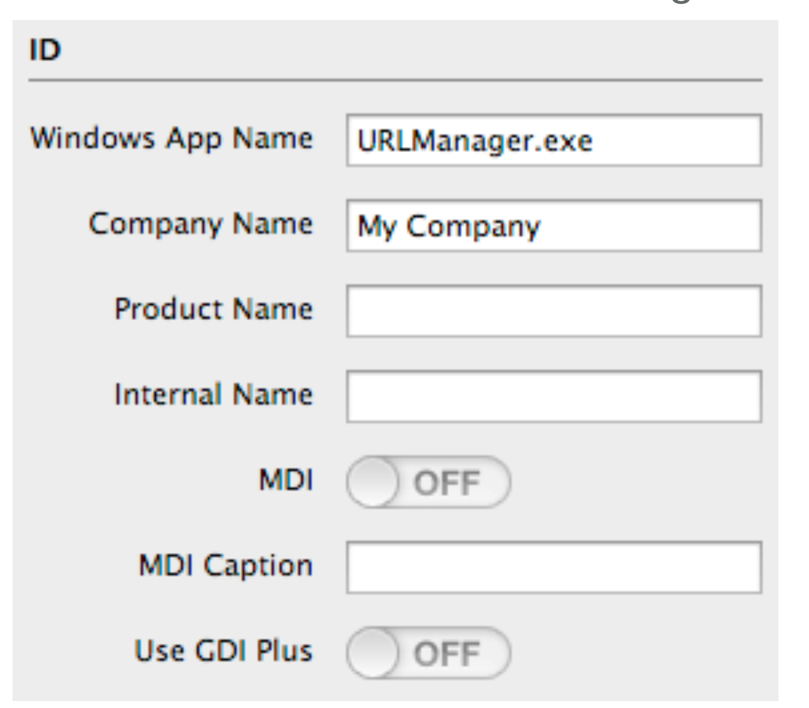
Abb. 4.4 OS X Build Einstellungen



Windows

Die Build-Einstellungen für Windows gestatten die Eingabe des Namens der Windows-Anwendung sowie den Unternehmensnamen und den internen Namen anzugeben, die in den Datei-Eigenschaften der Applikation angezeigt werden. Sie können auswählen, ob Sie ein Multiple-Document-Interface verwenden wollen und für diesen Fall seinen Titel festlegen und GDI Plus einschalten, das verbesserte Grafikinterface (enhanced Graphics Device Interface).

Abb. 4.5 Windows Build Einstellungen



Linux

In den Build-Einstellungen für Linux können Sie nur den Anwendungsnamen eingeben.

This Computer (Dieser Computer)

In diesem Abschnitt werden die Build-Einstellungen der verwendeten Entwicklungsplattform angezeigt. Wenn Sie z.B. Auf Mac OS X entwickeln, sehen Sie hier die OS X-Build-Einstellungen.

Build Ihrer Applikation

Sobald Sie die gewünschten Zielplattformen ausgewählt und die Build-Einstellungen vorgenommen haben, können Sie einen Build Ihrer Anwendung erzeugen lassen.
Klicken Sie dazu auf das Build-Symbol in der Werkzeugleiste (oder wählen Sie Project ↠ Build Application aus dem Menü).
Xojo wird eine Stand-Alone-Applikation für jede gewählte Plattform erzeugen.

Test Ihres Builds

Im Projektordner auf Ihrem Datenspeicher werden Sie nun ein Verzeichnis „Builds - DesktopTutorial.xojo_binary_project“ finden, in dem wiederum Ordner für die Builds jeder Plattform angelegt wurden.

Öffnen Sie das Verzeichnis Ihrer Plattform und klicken Sie auf das Programmsymbol, um Ihre Stand-Alone-App zu starten.

Abschnitt 4

Fertig!

Herzlichen Glückwunsch!

Sie haben das Desktop-Tutorial erfolgreich absolviert und eine voll funktionsfähige Anwendung erstellt.

Als nächsten Schritt auf Ihrer Xojo-Entdeckungsreise empfehlen wir Ihnen die Lektüre des Benutzerhandbuchs, das Xojo in seiner Gänze abdeckt.

Ziehen Sie ebenfalls die Sprachreferenz zurate, die die Details der Sprachelemente, Klassen und anderer Xojo-Features schnell nachschlagbar auflistet.

Action-Event

Wenn ein Benutzer auf einen PushButton klickt, wird der Programmcode im Action-Event-Handler ausgeführt.

Verwandte Glossarbegriffe

Drag related terms here

Index

Find Term

AddRow

Xojo-Programmierbefehl (Methode des Listbox-Steuerelements), um eine Zeile einer Listbox hinzuzufügen.

Verwandte Glossarbegriffe

List Box, Methode, Scope

Index Find Term

Anwendungs-ID

siehe Application Identifier

Verwandte Glossarbegriffe

Application Identifier

Index Find Term

Anwendungsname

siehe Application Identifier

Verwandte Glossarbegriffe

Application Identifier

Index Find Term

Application Identifier

Application Identifier stellt eine eindeutige Kennung Ihrer Anwendung dar. Aus den Eingaben der Felder Application Name und Company Name wird eine Vorgabe für dieses Feld erzeugt. Sie können Sie aber beliebig verändern.

Verwandte Glossarbegriffe

Anwendungs-ID, Anwendungsname

Index

Find Term

Application Name

Application Name ist der Name Ihrer Anwendung. Der hier eingegebene Begriff bestimmt den Dateinamen Ihrer Anwendung.

Verwandte Glossarbegriffe

Drag related terms here

Index

Find Term

Arbeitsbereich

siehe Workspace

Verwandte Glossarbegriffe

Workspace

Index Find Term

Ausnahmebehandlung

Siehe Exception

Verwandte Glossarbegriffe

Exception

Index Find Term

Bibliothek

siehe Library

Verwandte Glossarbegriffe

Library

Index Find Term

Boolesch

Eine boolesche Variable kennt nur zwei Zustände:
True (Wahr) oder False (Falsch).

Verwandte Glossarbegriffe

Enabled (aktiviert), False, True

Index Find Term

Build

Als Building wird die Umwandlung von Programm-Quellcode in ausführbaren Maschinencode bezeichnet, wobei der Quellcode in der Regel kompiliert, also in Maschinencode übersetzt wird und mit den nötigen Bibliotheken und weiteren Daten verlinkt. Es können sich weitere Schritte wie Optimierungen anschließen.

Ein Build ist entsprechend das Ergebnis eines solchen Prozesses: Eine ausführbare Anwendung.

Verwandte Glossarbegriffe

Kompilierung, Shared Build Settings, Stand-Alone-App

Index Find Term

Button

Ein normaler Bildschirmknopf. Häufig benutzt, um eine Programmaktion einzuleiten.

Verwandte Glossarbegriffe

Caption, Schaltfläche

Index Find Term

Kapitel 1 - Die Tutorial-Anwendung

Caption

= Titel, Beschriftung.
Eine Property vieler Benutzerinterface-Steuerelemente, deren Wert z.B. den angezeigten Text eines Buttons bestimmt.

Verwandte Glossarbegriffe

Button, Property

Index Find Term

Change-Event

Der Change-Event-Handler einer Listbox wird u.a. immer dann aufgerufen, wenn der Benutzer eine neue Zeile auswählt.

Verwandte Glossarbegriffe

Event (Ereignis), Event-Handler

Index Find Term

Code-Editor

Im Code-Editor ergänzen Sie Elemente Ihres Projekts mit Programmcode, z.B. In der Form von Event-Handlern oder Methoden.

Verwandte Glossarbegriffe

Drag related terms here

Index

Find Term

Company Name

Company Name ist der Name Ihres Unternehmens. Sie können dieses Feld auch leer lassen.

Verwandte Glossarbegriffe

Firmenname

Index Find Term

Debugger

Der Debugger ist der Teil eines Entwicklungssystems, der zur Verfolgung von Bugs benutzt werden kann. Sie haben hier die Möglichkeit, zur Laufzeit Veränderungen der Properties nachzuverfolgen und das Programm Schritt für Schritt abarbeiten zu lassen.

Verwandte Glossarbegriffe

Debugging

Index Find Term

Debugging

Als Debugging (wörtlich „Entwanzen“) bezeichnet man die Beseitigung von Programmfehlern, etwa um ungültige Benutzereingaben aufzufangen, die sonst zu einem Laufzeitfehler oder Fehlberechnungen führen könnten, oder einfach das Korrigieren falscher Methoden.

Verwandte Glossarbegriffe

Debugger, Laufzeitfehler, Methode

Index Find Term

Default Button

Der Default Button ist das Xojo-Steuerelement für eine Schaltfläche, die die bevorzugte Wahl darstellt, wie etwa ein OK-Knopf. Dieser Button ist via Betriebssystem auch per Tastaturkürzel (Zeilenschaltung, Return) ansprechbar.

Verwandte Glossarbegriffe

Generic Button

Index Find Term

Kapitel 2 - Schaltflächen (Buttons)

Dot notation

Siehe Punktnotation

Verwandte Glossarbegriffe

Punktnotation

Index Find Term

Eigenschaft

Siehe Property

Verwandte Glossarbegriffe

Property

Index Find Term

Kapitel 2 - Eigenschaften (Properties)

Enabled (aktiviert)

Eine Eigenschaft vieler Steuerelemente.
Ist Enabled wahr, so ist das Steuerelement ansprechbar.
Ist die Eigenschaft auf falsch gesetzt, so werden viele Elemente je nach Betriebssystem grau oder heller dargstellt und sich nicht anwählbar.

Verwandte Glossarbegriffe

Boolesch, False, Property, True

Index Find Term

Kapitel 4 - Verbesserungen

Event (Ereignis)

Bei der ereignisorientierten Programmierung wird ein Programm nicht linear durchlaufen. Stattdessen werden Methoden aufgerufen, wenn ein bestimmtes Ereignis auftritt. Diese Methoden werden in Xojo im Event-Handler definiert.

Verwandte Glossarbegriffe

Change-Event, Event-Handler, TextChange-Event

Index Find Term

Event-Handler

Ein Event-Handler dient in Xojo zur Aufnahme des Programmcodes einer Methode, die bei Auslösung eines bestimmten Events aufgerufen wird.

Verwandte Glossarbegriffe

Change-Event, Event (Ereignis), TextChange-Event

Index Find Term

Exception

Eine Exception (Ausnahme oder Ausnahmesituation) bezeichnet ein Verfahren, bei bestimmten Fehlerzuständen innerhalb eines Programms die normale Programmfortführung zu stoppen und z.B. Eine Methode zur Fehleranzeige oder -Behebung auszuführen.

Verwandte Glossarbegriffe

Ausnahmebehandlung, Methode, OutOfBoundsException

Index Find Term

Falsch

Siehe False

Verwandte Glossarbegriffe

False

False

Eine Zustandsmöglichkeit einer Booleschen Variable oder Eigenschaft.
Die andere ist True = Wahr.

Verwandte Glossarbegriffe

Boolesch, Enabled (aktiviert), Falsch, True

Index Find Term

Firmenname

siehe Company Name

Verwandte Glossarbegriffe

Company Name

Index Find Term

Gemeinsame Einstellungen

Siehe Shared Buld Settings

Verwandte Glossarbegriffe

Shared Build Settings

Index Find Term

Generic Button

Ein Generic Button (allgemeine Schaltfläche) ist das Xojo-Steuerelement für einen normalen Bildschirmknopf.

Verwandte Glossarbegriffe

Default Button

Index Find Term

Kapitel 2 - Schaltflächen (Buttons)

Global

Eine globale Variable oder Methode steht dem ganzen Projekt zur Verfügung.
Siehe auch Scope.

Verwandte Glossarbegriffe

Scope, ShowURL

Index Find Term

Kapitel 3 - Anzeigen-Schaltfläche

Http://

Eine URL wird üblicherweise mit „http://“ eingeleitet, womit angegeben wird, dass es sich bei der folgenden Adresse um eine Webseite handelt, die angezeigt werden soll.

Verwandte Glossarbegriffe

URL

Inspector

Der Inspector liefert Informationen über das im Navigator oder Editor ausgewählte Element.
Das Aussehen des Inspectors wird dem gerade gewählten Element angepasst.
Sie verändern eine Eigenschaft des angezeigten Elements, indem Sie einen Wert in das Texteingabefeld rechts der Feldbezeichnung eingeben.

Anzeige-Tastaturkürzel :
⌘-I auf OS X
STRG-I auf Windows und Linux

Verwandte Glossarbegriffe

Drag related terms here

Index

Find Term

Instanz (Instance)

Eine Instanz ist eine Arbeitskopie einer Klasse innerhalb eines Projekts.

So existiert z. B. eine Vielzahl eingebauter Klassen für Steuerelemente wie Schaltflächen, Web-Buttons, Etiketten, Textfelder usw..
Diese Steuerelemente sind aus sich heraus, als in der Library, nicht benutzbar. Sie stellen Matrizen für verwendbare Elemente dar. Wann immer Sie ein Steuerlement aus der Library in das Hauptfenster ziehen, wird eine Instanz dieser Klasse erzeugt.
So kann die Klasse eines Buttons mehrfach in einem Fenster verwendet werden: Jede Schaltfläche wird zu einer eigenständigen Instanz der Klasse, die unabhängig von den anderen Instanzen agiert.

Mit diesen Instanzen interagieren sowohl Sie, wenn Sie Ihre Funktionalität innerhalb eines Fensters programmieren, auf dem Sie platziert wurden, als auch später der Benutzer innerhalb der Anwendung.

Verwandte Glossarbegriffe

Klasse (Class)

Index

Find Term

Kapitel 2 - Eigenschaften (Properties)

Klasse (Class)

In ihrer einfachsten Form ist eine Klasse ein Behälter für Code wie Properties, Methoden und Steuerelemente. Klassen sind die grundlegenden Bausteine der objektorientierten Programmierung.
Eine Klasse existiert in einem Projekt nur einmal. Sie kann aber mehrfach innerhalb dessen instantiiert werden.

Verwandte Glossarbegriffe

Instanz (Instance), Property

Index Find Term

Kompilierung

Die Übersetzung eines Quellcodes in ausführbaren Maschinencode. Die Kompilierung ist Teil des Build-Vorgangs, beide Begriffe werden aber auch gerne synonym verwendet.

Verwandte Glossarbegriffe

Build

Index Find Term

Laufzeitfehler

Ein Programmfehler, der bei der Ausführung des Programms auftritt, etwa durch ungültige und nicht abgefangene Benutzereingaben.

Verwandte Glossarbegriffe

Debugging, Runtime Error

Layout-Editor

Layout-Editor: Das mittlere Gebiet des Workspace ist der Layout-Editor. Mit diesem erstellen Sie das Benutzerinterface für die Fenster Ihrer Anwendung. Der Layout-Editor liefert eine Vorschau auf das Erscheinungsbild Ihres Programmfensters.

Verwandte Glossarbegriffe

Drag related terms here

Index

Find Term

Library

Das Gebiet am rechten Rand des Hauptfensters stellt die Library dar, in der Sie die Steuer- und Interface-Elemente finden, die Sie zu einem Fenster oder fem Projekt hinzufügen können. Sie gestalten Ihr Fenster, indem Sie die Elemente aus der Library auf dieses hinüberziehen, oder indem Sie auf ein Element in der Library doppelklicken.

Anzeige-Tastaturkürzel :
⌘-L auf OS X
STRG-K auf Windows und Linux

Verwandte Glossarbegriffe

Bibliothek

Index Find Term

List Box

Ein Steuerelement, das eine scrollbare Liste darstellt. Dies kann sowohl eine einfache Liste als auch eine mehrspaltige Tabelle sein. Scrollen ist sowohl vertikal als auch horizontal möglich.

Verwandte Glossarbegriffe

AddRow, Listenelement, ListIndex, RemoveRow, Text

Index

Find Term

Listenelement

auch Liste, Listensteuerelement:
siehe List Box

Verwandte Glossarbegriffe

List Box

Index Find Term

Kapitel 1 - Die Tutorial-Anwendung

ListIndex

Property des Listbox-Steuerelements, in der die Nummer der ausgewählten Zeile der Listbox angeben wird.

Verwandte Glossarbegriffe

List Box, Property, Scope

Index

Find Term

Locking

Mithilfe der Schloss-Symbole in der Locking-Gruppe des Inspectors wird festgelegt, welche Positionen eines Elements bei der Skalierung seines Elternelements mitskaliert werden.

Verwandte Glossarbegriffe

Verriegelungspositionen

Index

Find Term

Mailto:

Steht am Anfang einer URL „mailto:“, wird die folgende Adresse stattdessen verwendet, um eine E-Mail über den voreingestellten Mail-Client zu versenden.

Verwandte Glossarbegriffe

URL

Index Find Term

Kapitel 1 - Die Tutorial-Anwendung

Me

Me bezieht sich auf das Steuerelement, das den aktuellen Event besitzt bzw. beinhaltet.

Außerhalb eines Event-Handlers bezieht sich Me auf das aktuelle Objekt.

Verwandte Glossarbegriffe

Drag related terms here

Index Find Term

Methode

Eine Methode besteht aus einem oder mehreren Programmiersprachbefehlen, die zusammen eine spezifische Aufgabe erledigen. Methoden sind die inneren Bausteine einer Anwendung. Der von Ihnen geschriebene Programmcode befindet sich hautpsächlich in Methoden.

Verwandte Glossarbegriffe

AddRow, Debugging, Exception, RemoveRow, ShowURL

Index Find Term

Navigator

Navigator: Das Panel am linken Bildschirmrand listet alle Bestandteile Ihres Projekts auf. Standardmäßig sehen Sie hier Window1 (das ausgewählt ist), das App-Objekt und das MainMenuBar-Objekt. Mittels des Navigators bewegen Sie sich innerhalb Ihres Projekts.

Verwandte Glossarbegriffe

Drag related terms here

Index Find Term

OutOfBoundsException

Eine OutOfBoundsException tritt auf, wenn ein verfügbarer Datenbereich überschritten wurde, etwa auf eine Zeile einer Listbox zugegriffen wird, die gar nicht existiert.

Verwandte Glossarbegriffe

Exception

Index Find Term

Kapitel 3 - Löschen-Schaltfläche

Projekt ausführen

Klicken Sie auf das Run-Symbol in der Werkzeugleiste, um es ausführen zu lassen.

Verwandte Glossarbegriffe

Drag related terms here

Index Find Term

Property

Eine Property ist ein Wert einer Klasse, wie etwa die Breite eines Fensters. Mittels Veränderung der Properties einer Klasse können Sie Verhalten und Aussehen einer Instanz der Klasse verändern.

Verwandte Glossarbegriffe

Caption, Eigenschaft, Enabled (aktiviert), Klasse (Class), ListIndex, Punktnotation, Text

---

Index

Find Term

Punktnotation

Eine häufig benutzte Schreibweise in objektorientierter Programmierung.
Wann immer Sie auf eine Property eines Objekts zugreifen wollen, verwenden Sie den Namen des Objekts, gefolgt von einem Punkt und dahinter den Namen der Eigenschaft.

ObjektName.PropertyName.

Verwandte Glossarbegriffe

Dot notation, Property

Index Find Term

RemoveRow

Xojo-Programmierbefehl (Methode des Listbox-Steuerelements), um eine Zeile einer Listbox zu löschen. Als Parameter wird die zu löschende Zeilennummer die die Methode übergeben.

Verwandte Glossarbegriffe

List Box, Methode, Scope

Index

Find Term

Runtime Error

Siehe Laufzeitfehler

Verwandte Glossarbegriffe

Laufzeitfehler

Index Find Term

Schaltfläche

auch Knopf:
siehe Button

Verwandte Glossarbegriffe

Button

Index Find Term

Kapitel 1 - Die Tutorial-Anwendung

Scope

Scope bezeichnet den Sichtbarkeitsbereich von z.B. Variablen und Methoden innerhalb eines Projekts. Der Scope legt fest, innerhalb welchen Bereichs das entsprechende Element sicht- und ansprechbar ist:

- **Public** (öffenlich): Das Element steht dem ganzen Projekt zur Verfügung.
- **Protected** (geschützt): Auf ein geschütztes Element kann nur innerhalb seiner Eltern-Klasse, seines Eltern-Fensters oder -Moduls zugegriffen werden. Die genauen Einschränkungen sind von der Art des Elternobjekts abhängig.
- **Private** (privat): Private Elemente können nur innerhalb der beinhaltenden Klasse oder des jeweiligen Moduls adressiert werden.

Verwandte Glossarbegriffe

AddRow, Global, ListIndex, RemoveRow, ShowURL, Sichtbarkeitsbereich

---

Index Find Term

Shared Build Settings

Die gemeinsamen Build-Einstellungen (Shared Build Settings) beinhalten die Versionsinformationen, Einstellungen für das Build-Verzeichnis und die Möglichkeit, zu Debugzweccken die Funktionsnamen in den Build zu integrieren.

Verwandte Glossarbegriffe

Build, Gemeinsame Einstellungen

Index Find Term

Kapitel 4 - Erzeugen einer Stand-Alone-App

ShowURL

Xojo-Programmierbefehl (globale Methode), um eine URL im voreingestellten Webbrowser des Benutzers zu öffnen.

Verwandte Glossarbegriffe

Global, Methode, Scope

Index Find Term

Sichern

Sie sollten Ihr Projekt regelmäßig speichern – und grundsätzlich vor dem Ausführen dessen.

Speichern Sie das Projekt via File ↠ Save.

Benennen Sie das Projekt „Schnellstart Desktop“ und klicken Sie auf Save.

Verwandte Glossarbegriffe

Drag related terms here

Index Find Term

Sichtbarkeitsbereich

Siehe Scope

Verwandte Glossarbegriffe

Scope

Index Find Term

Stand-Alone-App

Eine eigenständig ohne die Entwicklungsumgebung lauffähige Anwendung, im Gegensatz zur Debug-App, die innerhalb Xojos Debugger läuft.

Verwandte Glossarbegriffe

Build

Index Find Term

Text

Eine Property vieler Steuerelemente.
So enthält Textfield.Text den im Textfield stehenden bzw. eingegebenen Text,
Listbox.Text dagegen den Text der ausgewählten Zeile einer Listbox.

Verwandte Glossarbegriffe

List Box, Property, Text Field

Index Find Term

Text Field

Ein Steuerelement, das die Anzeige oder Eingabe einer Textzeile ermöglicht.

Verwandte Glossarbegriffe

Text, Textfeld

Index Find Term

Kapitel 1 - Die Tutorial-Anwendung

TextChange-Event

Der TextChange-Event-Handler eines Textfields wird bei jeder Änderung der Text-Property aufgerufen, sei diese durch Benutzereingabe oder durch Programmbefehle verursacht.

Verwandte Glossarbegriffe

Event (Ereignis), Event-Handler

Index Find Term

Kapitel 4 - Verbesserungen

Textfeld

siehe Text Field

Verwandte Glossarbegriffe

Text Field

Index

Find Term

True

Eine Zustandsmöglichkeit einer Booleschen Variable oder Eigenschaft.
Die andere ist False = Falsch.

Verwandte Glossarbegriffe

Boolesch, Enabled (aktiviert), False, Wahr

Index Find Term

Kapitel 4 - Verbesserungen

URL

Eine URL wird üblicherweise mit „http://“ eingeleitet, womit angegeben wird, dass es sich bei der folgenden Adresse um eine Webseite handelt, die angezeigt werden soll.

Steht am Anfang einer URL „mailto:“, wird die folgende Adresse stattdessen verwendet, um eine E-Mail über den voreingestellten Mail-Client zu versenden.

Verwandte Glossarbegriffe

Http://, Mailto:

Index Find Term

Verriegelungspositionen

Siehe Locking

Verwandte Glossarbegriffe

Locking

Index Find Term

Wahr

Siehe True

Verwandte Glossarbegriffe

True

Index Find Term

Workspace

Der Workspace ist das Hauptfenster von Xojo, in dem Sie Ihre Anwendung erstellen.

Verwandte Glossarbegriffe

Arbeitsbereich

Index

Find Term

Xojo Starten

- Laden Sie die Installationsdatei für Ihr Betriebssystem hier herunter: http://www.xojo.com/download
- Führen Sie das Installationsprogramm aus.
- Starten Sie Xojo.

Verwandte Glossarbegriffe

Drag related terms here

Index

Find Term